Consider the two languages *A* and *B*. The language *perfect shuffle* on *A* and *B* is as follows:
$\{w \mid w = a_1 b_1 \dots a_k b_k, \text{ where } a_1 \dots a_k \in A \text{ and } b_1 \dots b_k \in B, \text{ each } a_i, b_i \in \Sigma\}$.

Assume, $DFA_A = (Q_A, \Sigma, \delta_A, S_A, F_A)$ and $DFA_B = (Q_B, \Sigma, \delta_B, S_B, F_B)$ be two DFAs that recognize *A* and *B* respectively. $DFA_{Perfect-shuffle} = (Q, \Sigma, \delta, S, F)$ recognizes the language perfect shuffle on *A* and *B*.

The DFA for perfect shuffle switches from DFA_A to DFA_B after each character is read and it tracks the current states of DFA_A and DFA_B. Each character should belong to DFA_A or DFA_B i.e., $a_i, b_i \in \Sigma$. For each character read, $DFA_{Perfect-shuffle}$ makes moves in the corresponding DFA (either DFA_A or DFA_B). After the whole string is read, if both DFA_A and DFA_B reaches to the final state, then the input string is accepted by $DFA_{Perfect-shuffle}$.

The $DFA_{Perfect-shuffle}$ is defined as follows:

- $Q = Q_A \times Q_B \times \{A, B\}$: set of all possible states of DFA_A and DFA_B which should match with $DFA_{Perfect-shuffle}$.
- The input alphabet for $DFA_{Perfect-shuffle}$ is Σ.
- $q = (q_A, q_B, A)$: q_A and q_B are the initial states for DFA_A and DFA_B respectively. $DFA_{Perfect-shuffle}$ starts with q_A in DFA_A, q_B in DFA_B and the next character should be read from DFA_A.
- $F = F_A \times F_B \times \{A\}$: F_A and F_B are the final states for DFA_A and DFA_B respectively. $DFA_{Perfect-shuffle}$ accepts if both DFA_A and DFA_B reaches to the final states and the next character should be read from DFA_A.
- The transition function δ is,

1. $\delta((m, n, A), a) = (\delta_A(m, a), n, B)$
2. $\delta((m, n, B), b) = (m, \delta_B(n, b), A)$

Consider, the current state of DFA_A is *m* and the current state of DFA_B is *n*. Change the current state of *A* to $\delta_A(m, a)$ if the next character is to be read from DFA_A when *a* is the next character. After the character is read, read the next character from DFA_B. Change the current state of *B* to $\delta_B(n, b)$ if the next character is to be read from DFA_B when *b* is the next character.

The language *L* is said to be regular if there exist an FA that recognizes the language *L*. Here, the $DFA_{Perfect-shuffle}$ is defined for the language *perfect shuffle*.

Therefore, the class of regular languages is closed under perfect shuffle.