31. We'll prove that the perfect shuffle of two regular languages A and B is regular by constructing a DFA that recognizes it. Let's say that A and B are recognized respectively by the DFAs $D_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ and $D_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$. We construct a DFA D_S with the following 5-tuple.

$$
\begin{aligned}
Q_S &= Q_A \times Q_B \times \{A, B\} \\
\Sigma_S &= \Sigma \\
\delta_S((q, q', A), c) &\to (\delta_A(q, c), q', B) \\
\delta_S((q, q', B), c) &\to (q, \delta_B(q', c), A) \\
q_S &= (q_A, q_B, A) \\
F_S &= F_A \times F_B \times \{A\}
\end{aligned}
$$

D_S tracks the progress of the input string through both D_A and D_B by keeping track of the states of these DFAs, and alternating between their transition function upon reading each symbol. D_S starts out tracking the start states of both DFAs, and expecting to effect the next transition on D_A. D_S reaches a final state upon reading the input string iff the input string has a form $a_1 b_1 a_2 b_2 \cdots a_k b_k$ where $a_1 a_2 \cdots a_k \in L(D_A)$ and $b_1 b_2 \cdots b_k \in L(D_B)$, because such and only such strings can take the state of D_S to some state in $F_A \times F_B \times \{A\}$.

This concludes the proof.