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 as follows.

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

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.

33. We'll prove that $DROPOUT(A)$ is a regular language by constructing an NFA that recognizes it. Since A is a regular language, some DFA $D = (Q, \Sigma, \delta, q_0, F)$ recognizes it. We construct NFA N as follows.

$$\begin{aligned}
Q_N &= Q \cup copy(Q) \\
\Sigma_N &= \Sigma \\
\delta_N(q, c) &\rightarrow \delta(q, c) \\
\delta_N(q, \epsilon) &\rightarrow copy(q) \\
\delta_N(copy(q), c) &\rightarrow copy(\delta(q, c)) \\
q_{0N} &= q_0 \\
F_N &= copy(F)
\end{aligned}$$

For every state q in Q, we'll create a copy state $copy(q)$. Call the set of all copy states $copy(Q)$. We'll augment the transition function as follows. Firstly, we'll add identical transitions within $copy(Q)$ to what existing in Q. Secondly, we'll add ϵ transitions from Q to $copy(Q)$. The latter makes skipping a symbol equivalent to transitioning from some state q to $copy(q)$. The idea is that once the NFA has skipped a symbol, it enters some state in $copy(Q)$ and stays within $copy(Q)$ till the very end, since there is no way to go back into Q from $copy(Q)$. And further that unless the NFA skips a symbol (through an ϵ transition from Q to $copy(Q)$), it won't enter a state in $copy(Q)$ and thus won't be able to accept the input because $F_N = copy(F) \subseteq copy(Q)$. Therefore, N recognizes $DROPOUT(A)$.

This concludes the proof.