Let $A = \{ x \in \{0,1\}^* : x \text{ does not have 3 consecutive 0s in it} \}$.

Build a DFA $M$ such that $L(M) = A$.

The complement of $A$, which I denote by $\overline{A}$ or $\overline{A}$ (text $\overline{A}$) is $\{ x \in \{0,1\}^* : x \text{ does have 3 consecutive 0s in it} \}$.

First design a "goal-oriented" DFA $M'$ so $L(M') = \overline{A}$.

![DFA Diagram]

To get the original DFA $M$ for $L(M) = A$, (complement) the accepting and rejecting states.
Theorem: Given any DFA \( M = (Q, \Sigma, \delta, s, F) \) accepting a language \( A \), we can build a DFA \( M' = (Q', \Sigma, \delta', s', F') \) such that \( L(M') = \bar{A} \).

Proof: Design \( M' \) by taking \( Q' = Q \), \( s' = s \), and \( \delta' = \delta \) [i.e. states and arcs don't change].

But define \( F' = Q \setminus F \). Then

\[
L(M') = \{ x \in \Sigma^* : M' \text{ on input } x \text{ ends up in a state in } F' \} \\
= \{ x \in \Sigma^* : M' \text{ on } x \text{ does not end up in a state in } F \} \\
= \{ x \in \Sigma^* : M'(x) \text{ ends up in a state in } F \} \\
= \bar{L(M)} \text{ since } M \text{ and } M' \text{ work the same.}
\]

\( \Box \)
Cage-sim Product Example:

\[ A = \{ \bar{X} : 000 \text{ is not a substring of } X \} \]

\[ B = \{ \bar{X} : 11 \text{ is not a substring of } X \} \]

\[ M_A : \]

\[ M_B : \]

Design a DFA \( M_C \) such that \( L(M_C) = L(M_A) \cap L(M_B) \).

**Theorem:** For any two languages \( A, B \) accepted by DFAs \( M_A \) and \( M_B \), we can build a DFA \( M_C \) such that

\[ L(M_C) = L(M_A) \cap L(M_B) \]

Since \( \cap \) and \( \sim \) generate all Boolean operations, we can get \( A \cup B, A \Delta B \), etc. too.
Extra - was intended but will resume on Tue.
(or in recitations)

Let $\lor$ be any Boolean operation: $\land$, $\lor$, etc.
Given DFAs $M_A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ and $M_B = (Q_B, \Sigma, \delta_B, s_B, F_B)$, define $M_C = (Q_C, \Sigma, \delta_C, s_C, F_C)$ where:

$Q_C = Q_A \times Q_B$ states form $q_C = (q_A, q_B)$

$s_C = (s_A, s_B)$

$\delta((q_A, q_B), c) = (\delta_A(q_A, c), \delta_B(q_B, c))$

Finally $F_C = \{ (q_A, q_B) : q_A \in F_A \lor q_B \in F_B \}$

Set $\land, \lor, \land, \lor, \land$, etc., dead states
Can shorten

Set $\land, \lor, \land, \lor, \land$, etc., unchar
Can shorten

So if $\land, \lor = \text{XOR}$ then $F_C = \{ (q_A, q_B) : q_A \in F_A \land q_B \in F_B \} \lor \{q_A : q_A \in F_A \}$
**Defn:** A computation by a DFA $M = (Q, \Sigma, \delta, s, F)$ on a string $x \in \Sigma^*$ is a sequence

$$(q_0, x_1, q_1, x_2, q_2, \ldots, q_{n-1}, x_n, q_n)$$

such that $q_0 = s$,

for every $i$, $0 \leq i \leq n$, $\delta(q_{i-1}, x_i) = q_i$.

The computation is accepting if $q_n \in F$.

$L(M) = \{ x \in \Sigma^* : M \text{ has an accepting computation on input } x \}$