Regular Languages

... Spoken by the common folk...
Definition

A language is a Regular Language iff some Finite State Machine recognizes it.
Intuition:

What would make a language NOT Regular?

**Answer:** It requires memory.

*(other than tracking state, that is)*
A Few Examples (not limited to):  

Non-Regular languages  
- $\theta^n1^n$  
- $\theta^m1^n, m > n$  
- $w = (\theta|1)^*$  
  where $\#_\theta(w) = \#_1(w)$  
- $ww$ or $ww^R$  

Regular Languages  
- Contains a substring  
- $0^*1^*0^+$  
- Any finite language  
- Binary numbers divisible by 3
Regular Operations

Let $A$ and $B$ be languages. We define the regular operations union, concatenation, and star as follows:

- **Union:** $A \cup B = \{ x \mid x \in A \text{ or } x \in B \}$.
- **Concatenation:** $A \circ B = \{ xy \mid x \in A \text{ and } y \in B \}$
- **Star:** $A^* = \{ x_1 x_2 \ldots x_k \mid k \geq 0 \text{ and each } x_i \in A \}$
Question:

Are Regular Languages **closed** under Union, Concatenation, and Star?
Theorem: Regular Languages are Closed Under Union

Intuition:
- In terms of sets, what does Union mean?
- How can this be applied to two FSMs?
- Can you build a machine that models this behavior?

Define:
- $Q_3 = Q_1 \times Q_2$
  or, $Q_3 = \{(p, q) \mid p \in Q_1 \text{ and } q \in Q_2\}$
- $\Sigma$ (no change needed)
- $q_3 = (q_1, q_2)$
- $F_3 = (F_1 \times Q_2) \cup (Q_1 \times F_2)$
  Note: NOT $(F_1 \times F_2)$
- $\delta_3((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$
  where $a \in \Sigma$ and $(p, q) \in Q_3$.

Construct $M_3 = M_1 \cup M_2$

What else would this proof need?