Theorem. A language A is regular if and only if there exists an NFA M such that $L(M) = A$.

Proof. The forward direction is trivial, since A regular means there is a DFA that recognizes it, and a DFA can be seen as an NFA rather immediately.

So we focus on the backward direction. Assume that A is a language that is recognized by an NFA $M = (Q, \Sigma, \Delta, q_0, F)$. Without loss of generality, we know we can take the NFA to have no ϵ transitions. To show A is regular, we need to construct a DFA $M' = (R, \Sigma, \delta, r_0, G)$ that recognizes A. (To distinguish states of M from states of M', we use r to range over states of M', and R to represent the set of all states.)

The DFA M' will simulate the NFA M, in the sense that when following symbols of a string in M', the path taken will somehow capture all the possible paths that can be taken in M.

Define $M' = (R, \Sigma, \delta, r_0, G)$ by taking:

$$\begin{aligned} R &= \{r \mid r \subseteq Q\} \\ \delta(r, a) &= \{q' \mid q' \in \Delta(q, a) \text{ for some } q \in r\} \\ &= \cup_{q \in r} \Delta(q, a) \\ r_0 &= \{q_0\} \\ G &= \{r \mid r \cap F \neq \varnothing\} \end{aligned}$$

We now need to verify that $L(M') = L(M)$, that is, that M' and M recognize the same language. In other words, we need to show that for every string w, M accepts w if and only if M' accepts w.

We do this by induction on the length of w (since we need to show something true for an infinite number of things). First, define some useful notation. If δ is the transition relation of a DFA, then $\delta^*(q, w)$ tells you which state you end up in if you follow all the symbols in w from state q, based on the transition δ. Formally, δ^* is defined inductively on the structure of a string:

$$\begin{aligned} \delta^*(q, \epsilon) &= q \\ \delta^*(q, w \cdot a) &= \delta(\delta^*(q, w), a), \end{aligned}$$

where $w \cdot a$ is the concatenation of string w and symbol a. It is not hard to show that a DFA $M = (Q, \Sigma, \delta, q_0, F)$ accepts w if and only if $\delta^*(q_0, w)$ is in F.

Similarly, if Δ is the transition relation of an NFA without ϵ transitions, we can define Δ^* that tell you which states you can end up in if you follow all the symbols in w from state q of the NFA, based on the transition Δ. As above:

$$\Delta^*(q, \epsilon) = \{q\}$$
$$\Delta^*(q, w \cdot a) = \cup_{q' \in \Delta^*(q,w)} \Delta(q', a)$$

Again, it is not hard to show that an NFA $M = (Q, \Sigma, \Delta, q_0, F)$ accepts w if and only $\Delta^*(q_0, w)$ has a state that appears in F.

Now, given our automata M and M' as defined above, we show that for all strings w, $\delta^*(r_0, w) = \Delta^*(q_0, w)$. For the base case $w = \epsilon$, since $r_0 = \{q_0\}$, we have $\delta^*(r_0, \epsilon) = \delta^*(\{q_0\}, \epsilon) = \{q_0\} = \Delta^*(q_0, \epsilon)$, as required. For the inductive case, assume the result is true for a string w, we need to show it is true for a string $w \cdot a$: By definition, $\delta^*(r_0, w \cdot a) = \delta(\delta^*(r_0, w), a)$. By the induction hypothesis, $\delta^*(r_0, w) = \Delta^*(q_0, w)$, and thus $\delta^*(r_0, w \cdot a) = \delta(\Delta^*(q_0, w), a)$. By the definition of δ, $\delta(\Delta^*(q_0, w), a) = \cup_{q \in \Delta^*(q_0,w)} \Delta(q, a)$, which is just $\Delta^*(q_0, w \cdot a)$, as required. This proves the statement.

Now, suppose that M accepts w, that is, $\Delta^*(q_0, w) \cap F \neq \varnothing$. By the above result, this is equivalent to $\delta^*(r_0, w) \cap F \neq \varnothing$, that is, $\delta^*(r_0, w) \in G$, and this is equivalent to M' accepting w. This establishes that M and M' accept the same strings, that is, recognize the same language.