Kleene's Theorem

Some preliminary stuff:

We can impose an ordering upon the states in any FA: We assume w.l.o.g. that each state in a FA is numbered from 1 to n where $|Q| = n$.

We define
$R_{pq}^{()} = \{x \in \Sigma^* | \widehat{\delta}(p, x) = q\}$
This is the set of strings that "move" M from state p to state q.

Given that FA's do not exhibit nondeterminism each input string represents a single path from the start state to some concluding state (hopefully the concluding state is also an accepting state). Given this notion of a *path*, we can also say that a given path goes through a state.

$x \in \Sigma^*$ represents a path through a FA M. x goes through state s if $x = yz, |y|, |z| > 0, \widehat{\delta}(p, y) = s$ and $\widehat{\delta}(s, z) = q$ (note: $\widehat{\delta}(p, x) = q$).

We also define:
$R_{pq}^{(J)} = \{x \in \Sigma^* | x$ corresponds to a path from p to q that goes through no state numbered higher than $J\}$.

Notice that the notion of going through a state does not count the starting and stoping states. A path can start at state 7, end at state 8, and only *go through* states 1, 2, and 4. In this case we say that this path from $p = 7$ to $q = 8$ goes through no state numbered higher than $J = 4$.

The path represented by ϵ or single elements of Σ do not go *through* any states.

Finally we observe that $R_{pq}^{(n)} = R_{pq}^{()}$. This is true since there is no state numbered higher than n is our FA.

In order to prove our theorem:

1. Show each set $R_{pq}^{(n)}$ is a regular language
2. This is equivalent to showing each set $R_{pq}^{()}$ is regular language.
3. The sets $R_{q_0 f_i}^{()}, f_i \in F$ are special cases of $R_{pq}^{()}$; hence each set $R_{q_0 f_i}^{()}$ is a regular language.
4. The union of a set of regular languages is itself a regular language (regular languages are closed under union).

Theorem 1
If $M = (Q, \Sigma, \delta, q_0, F)$ is a FA recognizing the language R ($R = L(M)$), there is a regular expression over Σ corresponding to R.
Proof: We prove the theorem by using induction to show that for every p, $p \geq 1$, and $q, q \leq n$ each set $R_{pq}^{(J)}$, with $0 \leq J \leq n$ is a regular language.

Basis step: For every p and q, the set $R_{pq}^{(0)}$ represents a regular language. There is no state numbered 0; hence there is no state less than 0. Any path from p to q, therefore goes through no state (remember we do not consider the states p and q). Each set $R_{pq}^{(0)}$ therefore corresponds to either a single input symbol (a path of one transition is the only type of path that goes through no other states), or ϵ.

$R_{pq}^{(0)} \subseteq \Sigma \cup \{\epsilon\}$. Since every finite language is regular; the set $R_{pq}^{(0)}$ is a regular language.

Induction Hypothesis: For $0 \leq k \leq (n-1)$ and for every p and q, with $p \geq 1$, and $q \leq n$, the set $R_{pq}^{(k)}$ represents a regular language.

Need to show: For every p and q, with $p \geq 1$, and $q \leq n$, the set $R_{pq}^{(k+1)}$ is a regular language.

A string s is a member of the set $R_{pq}^{(k+1)}$ if x represents a path from p to q that goes through no state numbered higher than $k+1$. There are two cases to consider:

1. x bypasses state $k+1$ completely (there is only one state labeled $k+1$). This implies that $x \in R_{pq}^{(k)}$
2. x goes from state p to state $k+1$, from state $k+1$ x can go to other (lower numbered) states always returning to state $k+1$ (i.e. looping back to state $k+1$ some finite number of times), finally x goes from state $k+1$ to state q. In this case we write $x = x_1 y x_2$ (i.e. there are three components to the finite string x).
   (a) $\widehat{\delta}(p, x_1) = k+1$
   (b) $\widehat{\delta}(k+1, y) = k+1$
   (c) $\widehat{\delta}(k+1, x_2) = q$

   It should be clear that $x_1 \in R_{pk+1}^{(k)}$ and $x_2 \in R_{k+1q}^{(k)}$ (i.e. arriving at state $k+1$ for the first time x only goes through states numbered no higher than k, and leaving state $k+1$ for the last time, string x goes to q passing through no states numbered higher than k).

   There are two cases to consider:

   (a) $y = \epsilon$. The set $R_{pq}^{(k+1)}$ represents a regular language (the concatenation of two regular strings).

(b) $y \neq \epsilon$. The path that the string x represents loops from state $k+1$ back to state $k+1$ one or more times. We can represent each looping from state $k+1$ back to state $k+1$ by some portion of the string y. $y = y_1 y_2 \ldots y_r$ Each y_i portion of the string y is an element of the set $R^{(k)}_{k+1k+1}$. Hence $y \in (R^{(k)}_{k+1k+1})^*$

We can now say

$$R^{(k+1)}_{pq} = R^{(k)}_{pq} \cup R^{(k)}_{pk+1}(R^{(k)}_{k+1k+1})^* R^{(k)}_{k+1q}$$

By the induction hypothesis each set on the right side represents a regular language (regular languages are closed under union, concatenation and *). Therefore the set $R^{(k+1)}_{pq}$ represents a regular language (completing the induction proof, and proving the theorem). $\diamond$

We have shown that

1. For each p and q and J, $0 \leq J \leq n$, the set $R^{(J)}_{pq}$ is a regular language.
2. $R^{()}_{pq} = R^{(n)}_{pq}$ Hence the set $R^{()}_{pq}$ is a regular language.
3. since $L(M) = \cup_{f_i \in F} R^{()}_{q0a_i}$, $L(M)$ is a regular language.