An NFA is a 5-tuple $N = (Q, \Sigma, \delta, s, F)$ where $Q, \Sigma, \delta, F$ are as before and:

$$S \subseteq Q \times \Sigma \times Q$$

**Example**

$$N_3 : \quad \xrightarrow{0,1} \quad 1 \quad \xrightarrow{0} \quad 0 \quad \xrightarrow{0} \quad f$$

$$L(N_3) = \{x \in \{0,1\}^* : \text{the 3rd char from the right is a 1}\}$$

A DFA "Is a" NFA in which the $S$ relation is actually a function from $Q \times \Sigma$ to $Q$.

**Def.** An NFA can process a string $x$ from state $p$ to state $q$ ($q \neq p$ allowed) if there is a computation

$$(q_0, x_1, q_1)(q_1, x_2, q_2) \cdots (q_{n-2}, x_n, q_{n-1})(q_{n-1}, x_n, q_n)$$

which is a sequence of instructions that "match like domino where $q_0 = p$, $q_n = q$, and $x = x_1 \cdots x_n$.

**Def.** An NFA also allows instructions $(p, e, q)$. The processing then now has $(q_0, u_1, q_1) \cdots (q_{t-1}, u_t, q_t)$ where $x = u_1 \cdots u_t$ and the new can begin when $u_t = 1$.
Example: $N_3$ can process $x = 1011$ from $s$ to $q_3$. Let's try some computation traces:

Too Eager:
$$(5,1,9_2)(9_2,0,9_3) \text{ must next do (9_3,1,f)}$$
but cannot process the final input $x$

"Too Timid":
$$(5,1,5)(5,0,5)(5,1,5)(5,1,9_2)$$
but fails short of $q = q_3$

"Just Right":
$$(5,1,5_1, (5,0,5)(5,1,9_2)(9_2,1,9_3). \checkmark$$

Defn: Given $N$, $L_{p,q} = \{ x \in \Sigma : N \text{ can process } x \text{ from } p \text{ to } q \}$
and now formally:
$$L(N) = \bigcup_{q \in F} L_{s,q}.$$

Note how the interpretation is "benefit of doubt":
$1011 \notin L_{s,q_3}$, even though your odds are $1/2^3$!

But $1011 \in L_{s,q}$, so not in $L(N)$, because there is no way to process it all from $s$ to $f$.

Regular Expression: $L(N_3) = (0+1)^*1(0+1)^2$. 

Defn. and Theorem with Regexps, by Induction.

**Basis:**
- $\emptyset$ is a regexp, $L(\emptyset) = \emptyset$, and $N_\emptyset = n_\emptyset$.
- $\varepsilon$ is a regexp, $L(\varepsilon) = \{\varepsilon\}$, and $N_\varepsilon = n_\varepsilon$.

For any char $c \in \Sigma$,
- $\langle c \rangle$ is a regexp, $L(\langle c \rangle) = \{c\}$, and $N_c = n_c$.

The language (set of strings) whose only member is the string “$c$”. Now $N_c$ cannot anymore process $\varepsilon$ from $S \not\rightarrow F$, so $\varepsilon \notin L(N_c)$, indeed, $L(N_c) = \{\varepsilon\}$.

**Induction:** Let any two regexps $\alpha$ and $\beta$ be given. Then
- $\varphi = \alpha + \beta$ is a regexp, $L(\varphi) = L(\alpha) \cup L(\beta)$.

And by inductive hypotheses we may take NFAs $N_\alpha$ and $N_\beta$ such that $L(N_\alpha) = L(\alpha)$ and $L(N_\beta) = L(\beta)$.

To be continued...