NFA with epsilon transitions

Sipser  pages 47-54
NFA’s with $\varepsilon$ – Transitions

• We extend the class of NFAs by allowing instantaneous ($\varepsilon$) transitions:
  1. The automaton may be allowed to change its state without reading the input symbol.
  2. In diagrams, such transitions are depicted by labeling the appropriate arcs with $\varepsilon$.
  3. Note that this does not mean that $\varepsilon$ has become an input symbol. On the contrary, we assume that the symbol $\varepsilon$ does not belong to any alphabet.
example

\[ \{ a^n \mid n \text{ is even or divisible by 3} \} \]
Definition

• An ε-NFA is a quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where

  - $Q$ is a set of states
  - $\Sigma$ is the alphabet of input symbols
  - $q_0 \in Q$ is the initial state
  - $F \subseteq Q$ is the set of final states
  - $\delta: Q \times \Sigma_\varepsilon \rightarrow P(Q)$ is the transition function

• Note $\varepsilon$ is never a member of $\Sigma$

• $\Sigma_\varepsilon$ is defined to be $(\Sigma \cup \varepsilon)$

This is the version of the NFA on page 53 of Sipser.
• $\varepsilon$-NFAs add a convenient feature but (in a sense) they bring us nothing new: they do not extend the class of languages that can be represented. Both NFAs and $\varepsilon$-NFAs recognize exactly the same languages.

• $\varepsilon$-transitions are a convenient feature: try to design an NFA for the even or divisible by 3 language that does not use them!
  – Hint, you need to use something like the product construction from union-closure of DFAs
ε-Closure

- ε-closure of a state
- The ε-closure of the state q, denoted ECLOSE(q), is the set that contains q, together with all states that can be reached starting at q by following only ε-transitions.

In the above example:
- ECLOSE(P) =\{P,Q,R,S\}
- ECLOSE(R) =\{R,S\}
- ECLOSE(x) =\{x\} for the remaining 5 states \{Q,Q1,R1,R2,R2\}
Computing $e\text{close}$

- Compute $e\text{close}$ by adding new states until no new states can be added
  - Start with $[P]$
  - Add $Q$ and $R$ to get $[P,Q,R]$  
  - Add $S$ to get $[P,Q,R,S]$  
  - No new states can be added
Elimination of \( \varepsilon \)-Transitions

• Given an \( \varepsilon \)-NFA \( N \), this construction produces an NFA \( N' \) such that \( L(N')=L(N) \).

• The construction of \( N' \) begins with \( N \) as input, and takes 3 steps:

1. Make \( p \) an accepting state of \( N' \) iff \( ECLOSE(p) \) contains an accepting state of \( N \).
2. Add an arc from \( p \) to \( q \) labeled \( a \) iff there is an arc labeled \( a \) in \( N \) from some state in \( ECLOSE(p) \) to \( q \).
3. Delete all arcs labeled \( \varepsilon \).
Make $p$ an accepting state of $N'$ iff $\text{ECLOSE}(p)$ contains an accepting state of $N$.

Add an arc from $p$ to $q$ labeled $a$ iff there is an arc labeled $a$ in $N$ from some state in $\text{ECLOSE}(p)$ to $q$.

Delete all arcs labeled $\epsilon$. 
Why does it work?

• The language accepted by the automaton is being preserved during the three steps of the construction: \( L(N) = L(N_1) = L(N_2) = L(N_3) \)

• Each step here requires a proof. A Good exercise for you to do!
Theorem

• Any NFAe can be turned into an NFA

• How?