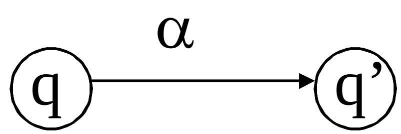
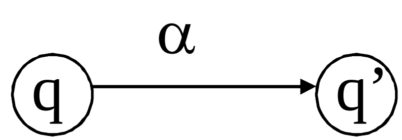
NFA vs DFA

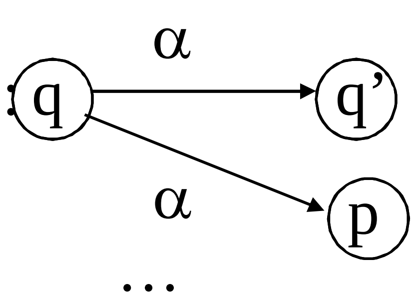
DFA: For every state q in S and every character α in Σ, one and only one transition of the following form occurs:



NFA: For every state q in S and every character α in $\Sigma \cup \{e\}$, one (or both) of the following will happen:

- No transition: 

occurs

- **One or more** transitions: 

occurs

NFA vs DFA (2)

All deterministic automata are non deterministic

Given a nondeterministic automaton, it is always possible to find a an equivalent deterministic automaton “doing the same”?

That is, given an NFA $M = (Q,\Sigma, \delta,s,F)$ does there exists an equivalent DFA $M' = (Q',\Sigma, \delta',s',F')$? **YES!**

δ: $\mathbf{Q} \times \mathbf{(}\Sigma \cup \{e\}\mathbf{)} \times \wp(Q)$ δ': $Q' \times \Sigma \rightarrow Q'$

We are going to construct the DFA by using the given NFA

Equivalence of NFA and DFA

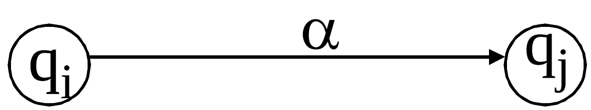
Definition. Two automata A and A’ are **equivalent** if they recognize the same language.

Theorem. Given any NFA A, then there exists a DFA A’ such that A’ is equivalent to A

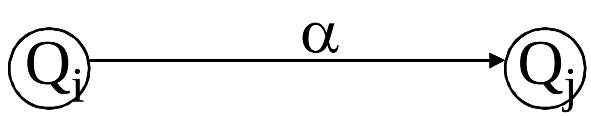
Idea of the Transformation: NFA → DFA

We would like:

For every transition in NFA:

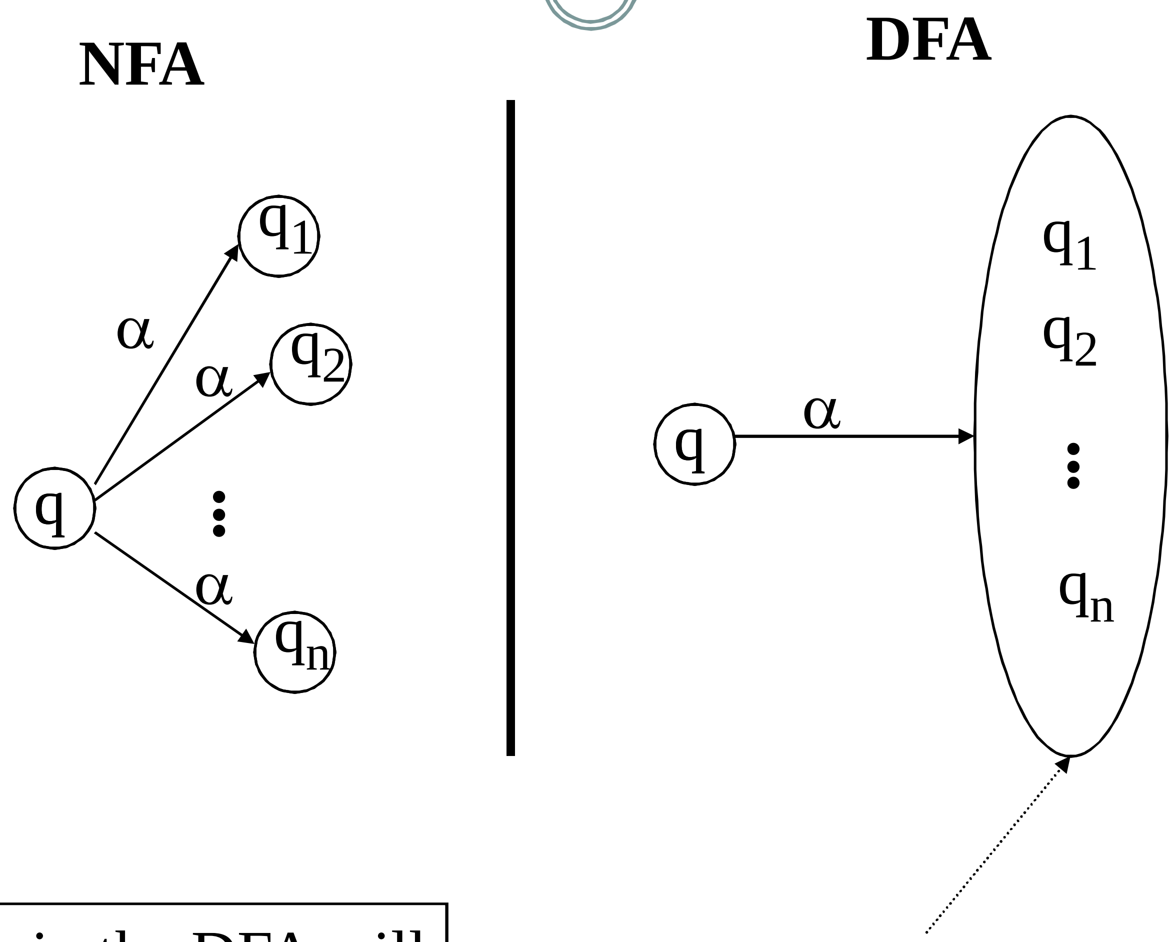


There is a transition in the equivalent DFA:



where Q_i (or Q_j) is related to q_i (or q_j)

Idea (2): Remove Non Determinism

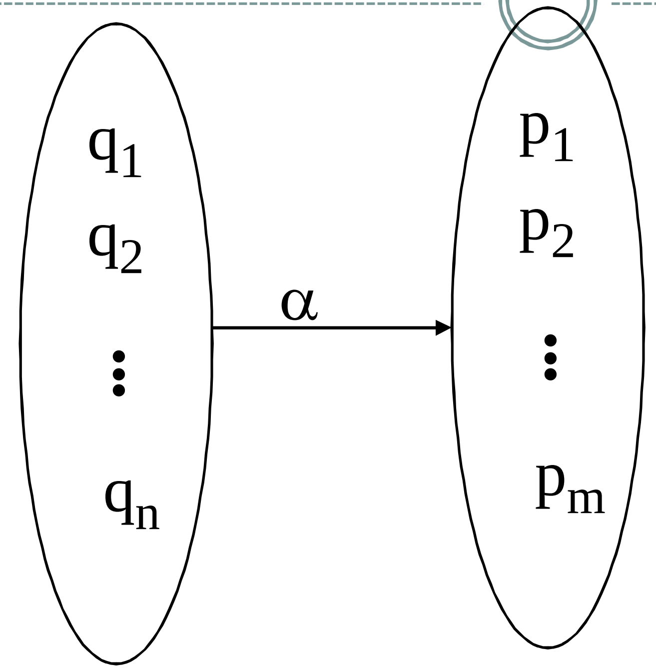


The states in the DFA will be elements in $\wp(Q)$

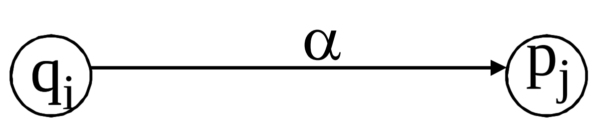
This is the set: $\delta(q, \alpha)$

Step 1: Assign Arcs

DFA:



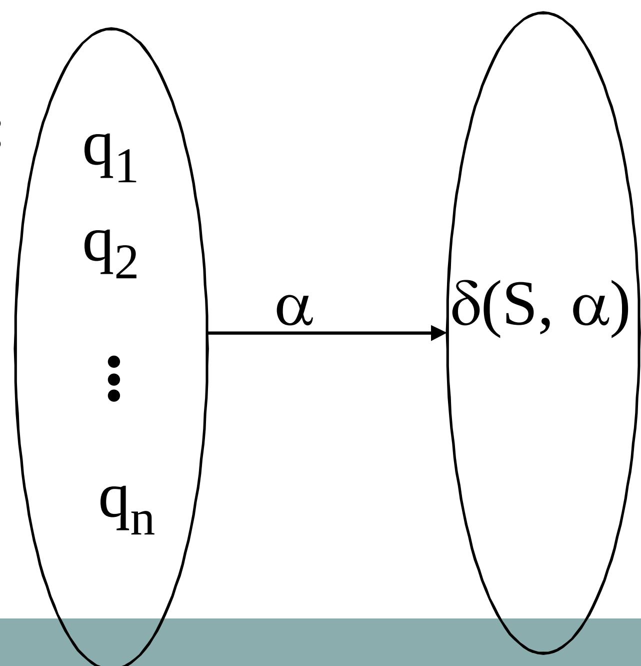
If in the original NFA:



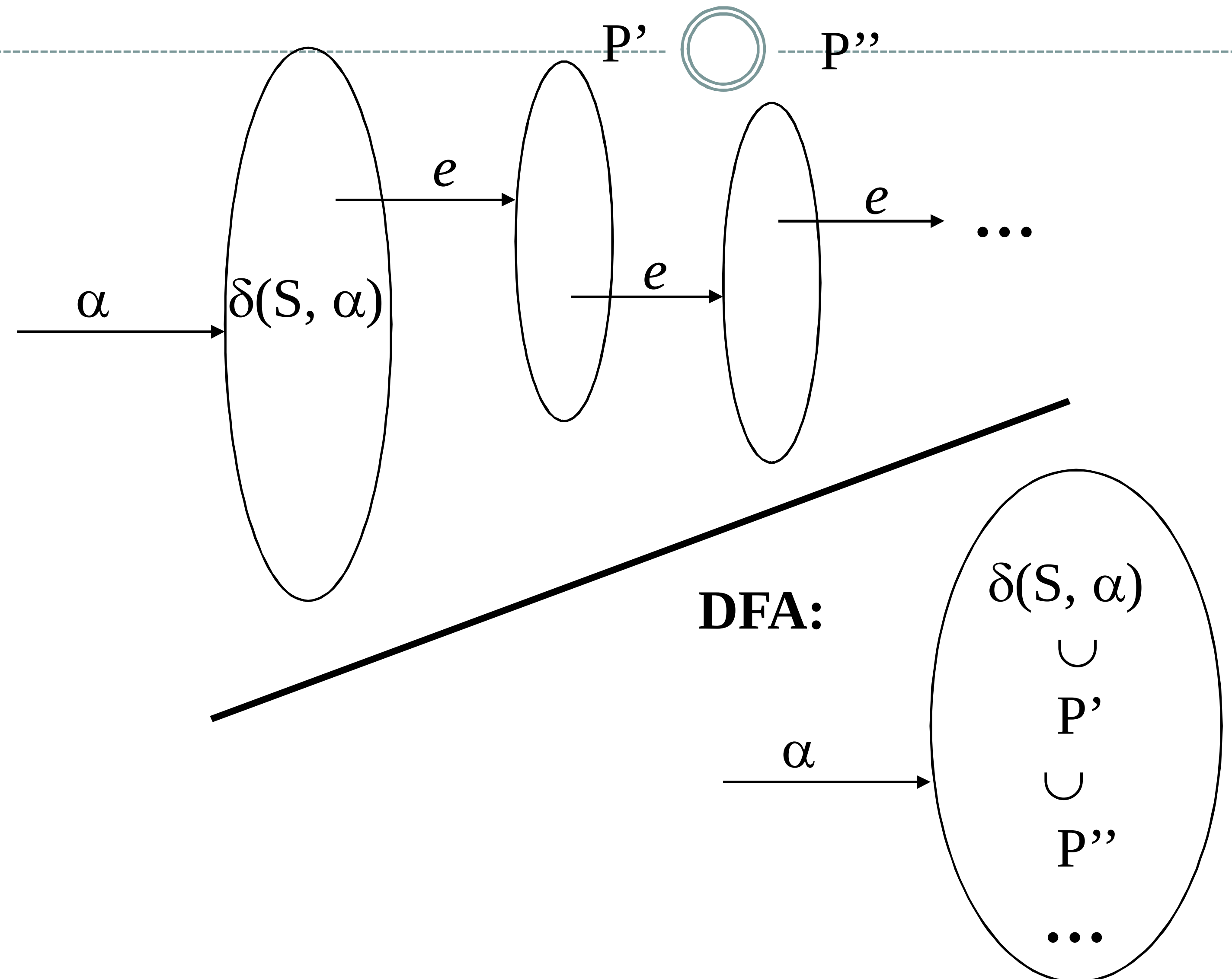
Step 1 : Variation

Let S be an state formed by $\{q_1, q_2, \ldots, q_n\}$, we denote the set $\delta(S, \alpha)$ as the set of all states that are reachable from states in S by reading α

DFA:



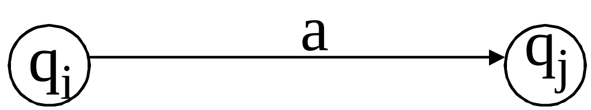
Step 2 : Eliminating *e*-Transitions



Step 3: Handling Undetermined Transitions

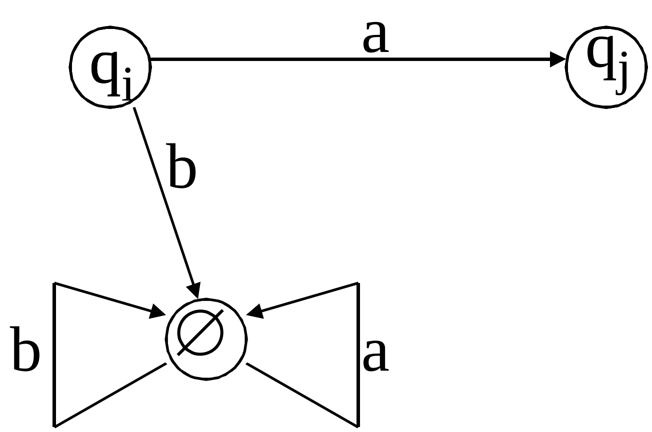
Suppose that Σ = { a, b} and we have only a transition for a:

NFA:



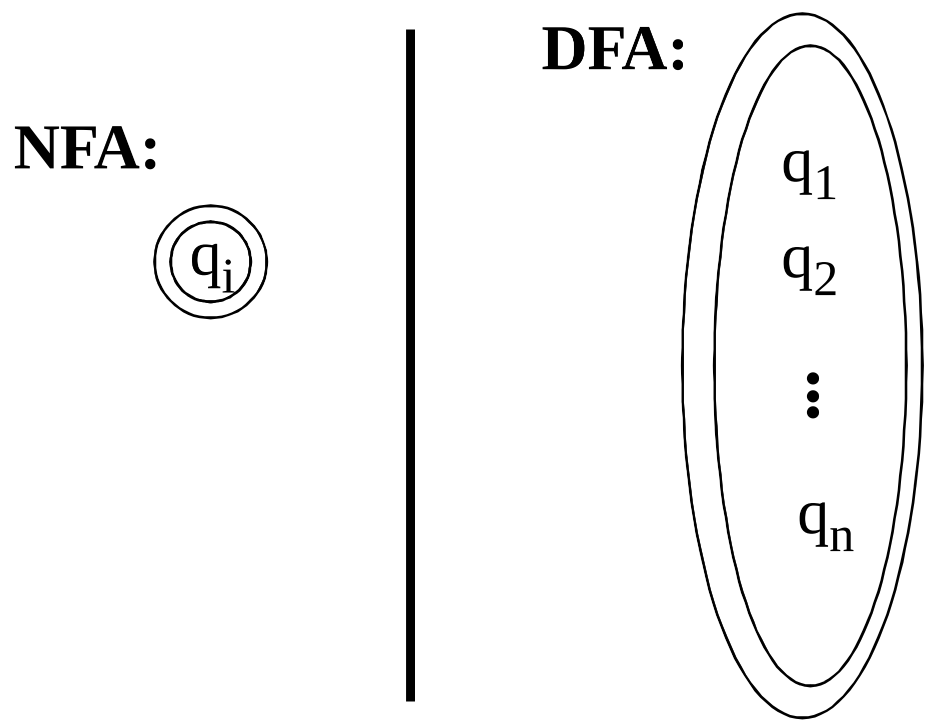
What should we do for b?

DFA:

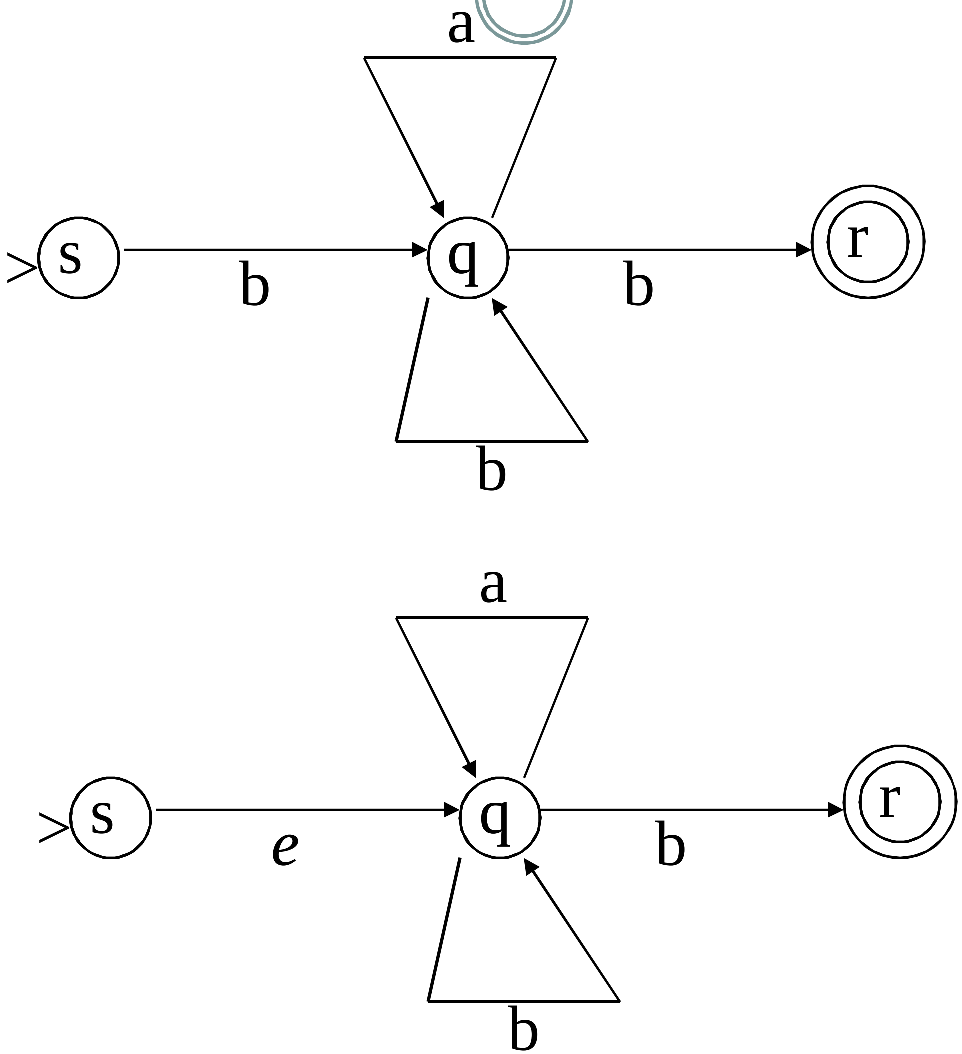


Step 4: Determining Favorable States

We will make states favorable in the DFA only if they contain at least one state which is favorable in the NFA



Examples



Proof

Given an NFA M = (Q,Σ,Δ,s,F) suppose that we use the procedure discussed to obtain a DFA M' = (Q',Σ, δ,s',F'). What needs to be shown to prove that M and M' are **equivalent**?

- For each w accepted by M', w is also accepted by the NFA
- For each w accepted by M, w is also accepted by the DFA

We will show the first one for a "generic" word:

$$w = \alpha_1 \alpha_2 \ldots \alpha_n$$

Where each α_i is in Σ

Proof (2)

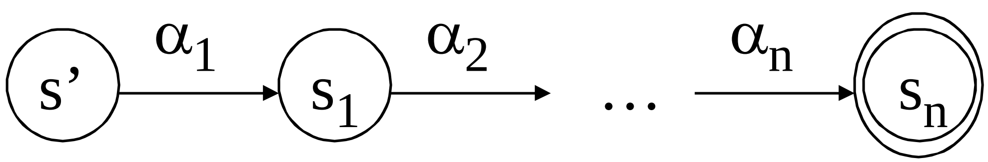
• Proof by induction on the length n of the word

$$w = \alpha_1\ \alpha_2 \ldots \alpha_n$$

- $n = 1$
- $n = k \Rightarrow n = k+1$

• Suppose that w is accepted by the DFA, what does this means?

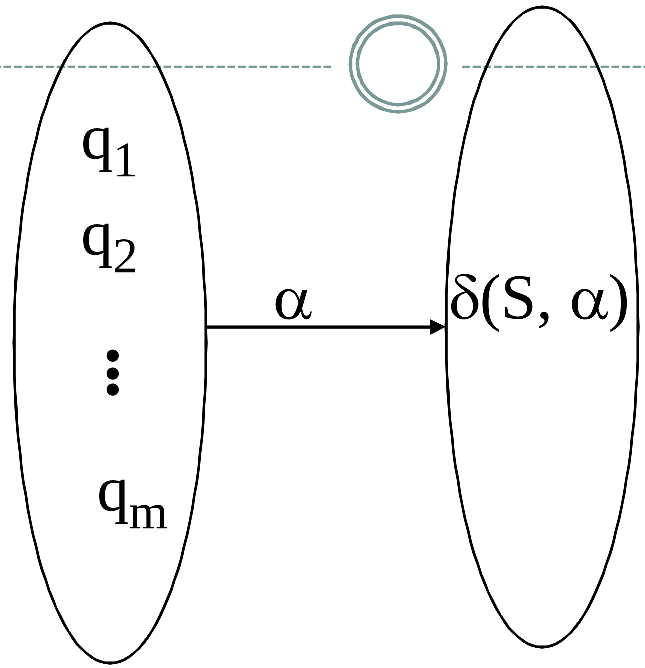
D:



Where s’ and each s_i and s’ are states in the DFA
(i.e., elements in $\wp$(Q); where Q are the states in the NFA)

Construction

D:

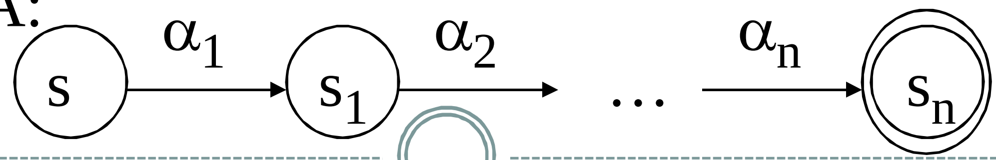


states in the NFA

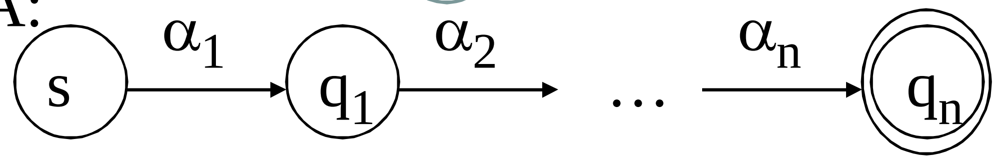
Where $S = \{q_1, q_2, \ldots, q_m\}$,

Assume no e-transitions for the moment

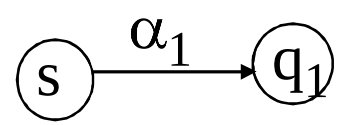
We have: DFA:



We want: NFA:

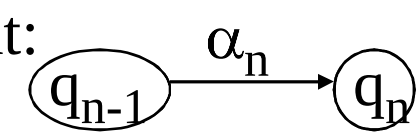


$s_1 = \delta(s, \alpha_1)$ → Select state q_1 in s_1 such that:



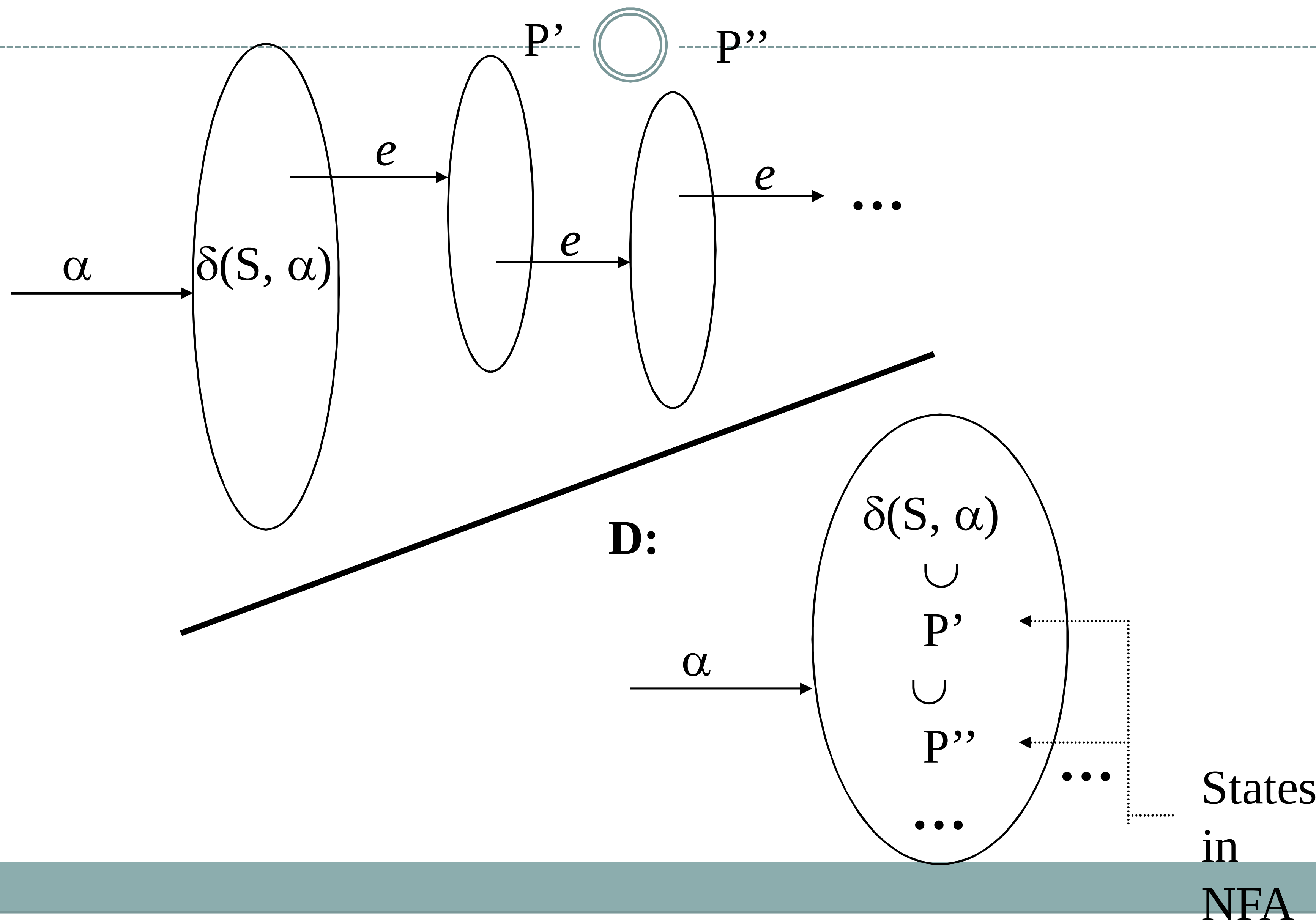
…

$s_{n-1} = \delta(s_{n-2}, \alpha_{n-1})$ → Select state q_{n-1} in s_{n-1} such that:

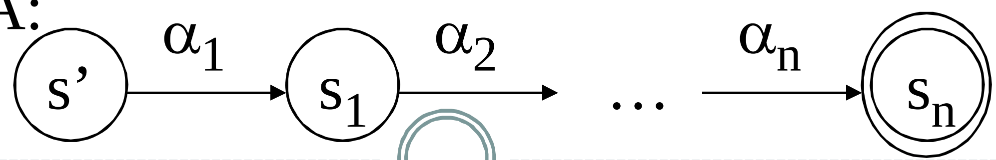


$s_n = \delta(s_{n-1}, \alpha_n)$ → Select state q_n in s_n such that q_n is favorable in DFA

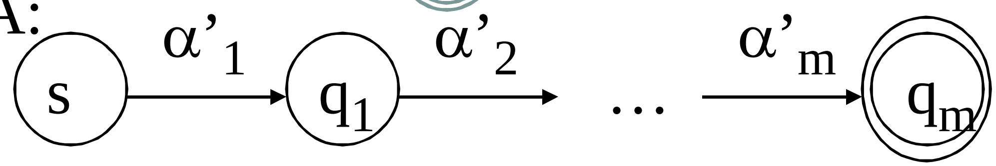
Dealing with e-transitions



We have: DFA:

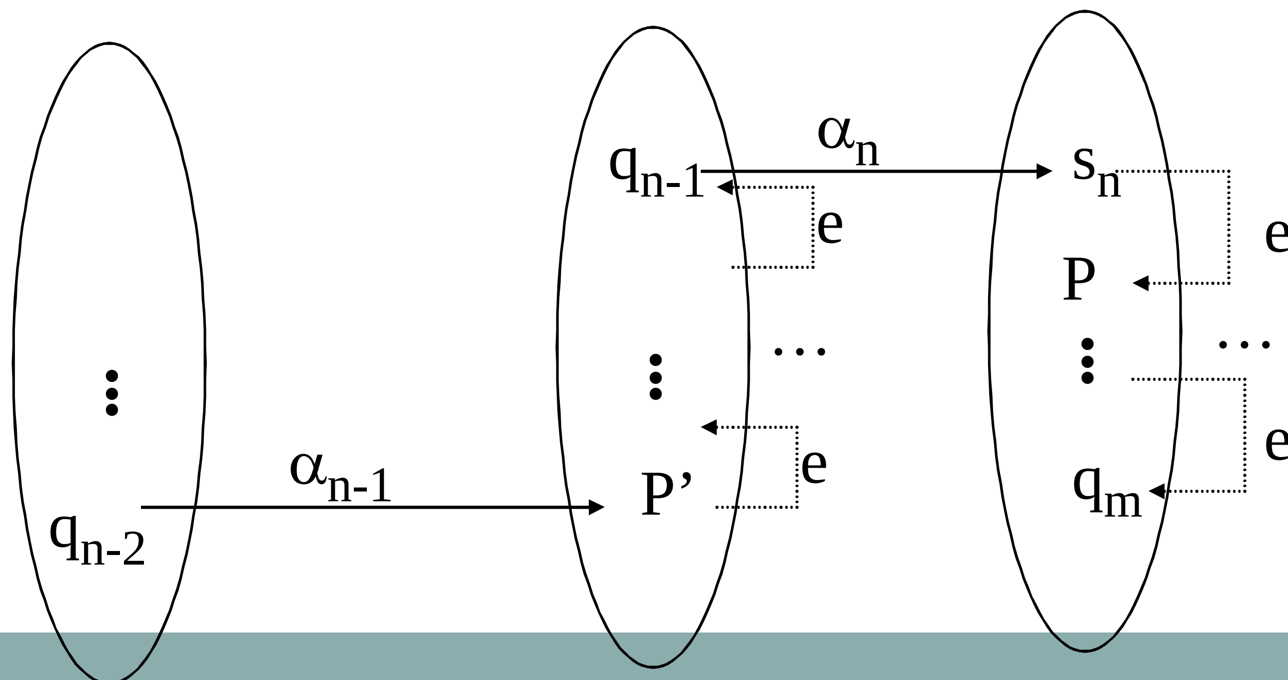


We want: NFA:



$m \geq n$

each α'_i is either an α_j or *e*



Main Result

The other direction is very simple (**do it!**):

For each w accepted by N, w is also accepted by D

Theorem. Given any NFA N, then there exists a DFA D such that N is equivalent to D