Equivalence of DFA and NFA

- Equivalence of DFA and NFA
- Method for reduction of NFA to DFA-, Subset-Construction
- Theorems for equivalence of Language accepted by DFA and NFA

#Hemanta GC

Equivalence of DFA's, NFA's

- A DFA can be turned into an NFA that accepts the same language.
- If for a DFA, $\delta_D(q, a) = p$, let the NFA have $\delta_N(q, a) = \{p\}$.
- Then the NFA is always in a set containing exactly one state as the state the DFA is in after reading the same input.

Equivalence

- Surprisingly, for any NFA there is a DFA that accepts the same language.
- The DFA has about as many states as NFA but it often has more transitions.
- In the worst case, the smallest DFA that is equivalent to some NFA with n states, can have 2^n states.
- NFA is easier to construct because it has minimum states than DFA and minimum transitions.
- After construction of NFA, it can be converted to DFA.

Subset Construction

- Given an NFA with states Q, inputs Σ, transition function δ_N, state state q_0, and final states F, construct equivalent DFA with:
  - States in 2^Q (Set of subsets of Q).
  - Inputs Σ.
  - Start state $\{q_0\}$.
  - Final states = all those in 2^Q with at least member of F.
  - Let S in 2^Q is one of the state of DFA so the transition function is $\delta_D(S,a) = \cup_{p \text{ in } S} \delta_N(p, a)$

Critical Point

- The DFA states have *names* that are sets of NFA states.
- But as a DFA state, an expression like {p,q} must be read as a single symbol, not as a set.
- Analogy: a class of objects whose values are sets of objects of another class.

Subset Construction

- The transition function δ_D is defined by:

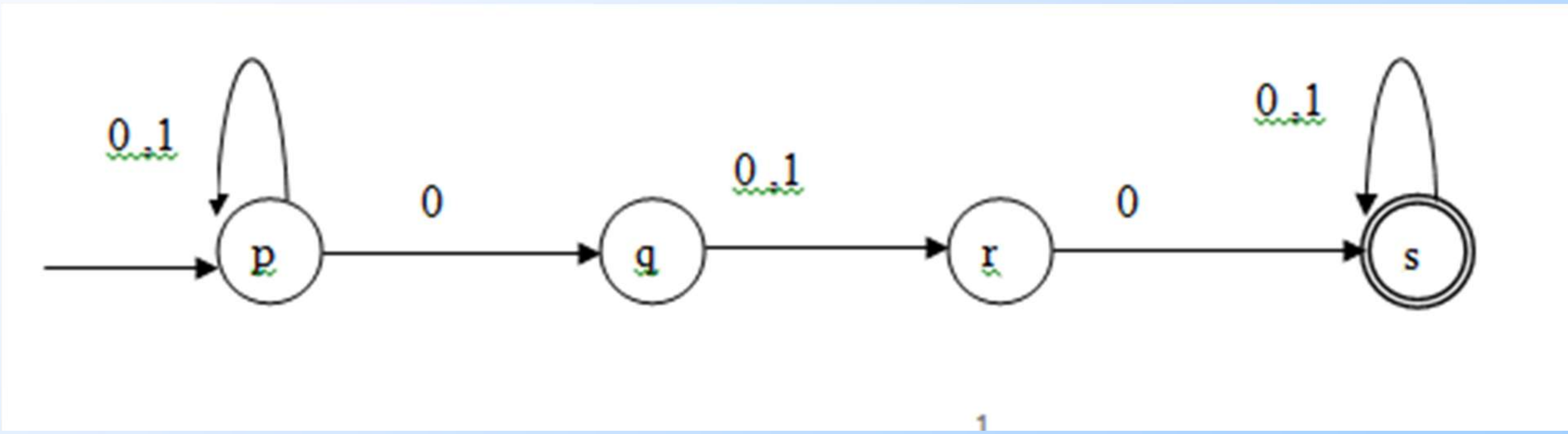
$\delta_D(\{q_1,\ldots,q_k\}, a)$ is the union over all $i = 1,\ldots,k$ of $\delta_N(q_i, a)$.

- If $S = \{q_1,\ldots,q_k\}$ then δ_D can be defined by $\delta_D(S,a) = \bigcup_{p \text{ in } S} \delta_N(p, a)$

- Example: We'll construct the DFA equivalent of our "chessboard" NFA.

Subset Construction- An Example:

- Convert the following NFA $N = (Q, \Sigma, \delta , q_0 , F)$ to DFA.



Example: Subset Construction

NFA

	0	1
→ p	{p,q}	{p}
q	{r}	{r}
r	{s}	{}
*s	{s}	{s}

Equivalent DFA

	0	1
→ {p}	{p,q}	{p}
{p,q}		

Example: Subset Construction

NFA

	0	1
→p	{p,q}	{p}
q	{r}	{r}
r	{s}	{}
*s	{s}	{s}

Equivalent DFA

	0	1
→ {p}	{p,q}	{p}
{p,q}	**{p,q,r}**	**{p,r}**
{p,q,r}		
{p,r}		

Example: Subset Construction

NFA

	0	**1**
→p	**{p,q}**	**{p}**
q	**{r}**	**{r}**
r	**{s}**	**{}**
***s**	**{s}**	**{s}**

Equivalent DFA

	0	1
→ {p}	{p,q}	{p}
{p,q}	{p,q,r}	{p,r}
{p,q,r}	**{p,q,r,s}**	**{p,r}**
{p,r}		
*{p,q,r,s}		

Example: Subset Construction

NFA

	0	1
→p	{p,q}	{p}
q	{r}	{r}
r	{s}	{}
*s	{s}	{s}

Equivalent DFA

	0	1
→ {p}	{p,q}	{p}
{p,q}	{p,q,r}	{p,r}
{p,q,r}	{p,q,r,s}	{p,r}
{p,r}	**{p,q,s}**	**{p}**
*{p,q,r,s}		
*{p,q,s}		

Example: Subset Construction

NFA

	0	1
→p	{p,q}	{p}
q	{r}	{r}
r	{s}	{}
*s	{s}	{s}

Equivalent DFA

	0	1
→ {p}	{p,q}	{p}
{p,q}	{p,q,r}	{p,r}
{p,q,r}	{p,q,r,s}	{p,r}
{p,r}	{p,q,s}	{p}
***{p,q,r,s}**	**{p,q,r,s}**	**{p,r,,s}**
*{p,q,s}		
*{p,r,s}		

Example: Subset Construction

NFA

	0	1
→p	**{p,q}**	**{p}**
q	**{r}**	**{r}**
r	**{s}**	**{}**
*s	**{s}**	**{s}**

Equivalent DFA

	0	1
→ {p}	{p,q}	{p}
{p,q}	{p,q,r}	{p,r}
{p,q,r}	{p,q,r,s}	{p,r}
{p,r}	{p,q,s}	{p}
*{p,q,r,s}	{p,q,r,s}	{p,r,,s}
***{p,q,s}**	**{p,q,r,s}**	**{p,r,s}**
*{p,r,s}		

Example: Subset Construction

NFA

	0	1
→p	{p,q}	{p}
q	{r}	{r}
r	{s}	{}
*s	{s}	{s}

Equivalent DFA

	0	1
→ {p}	{p,q}	{p}
{p,q}	{p,q,r}	{p,r}
{p,q,r}	{p,q,r,s}	{p,r}
{p,r}	{p,q,s}	{p}
*{p,q,r,s}	{p,q,r,s}	{p,r,,s}
*{p,q,s}	{p,q,r,s}	{p,r,s}
***{p,r,s}**	**{p,q,s}**	**{p,s}**
*{p,s}		

Example: Subset Construction

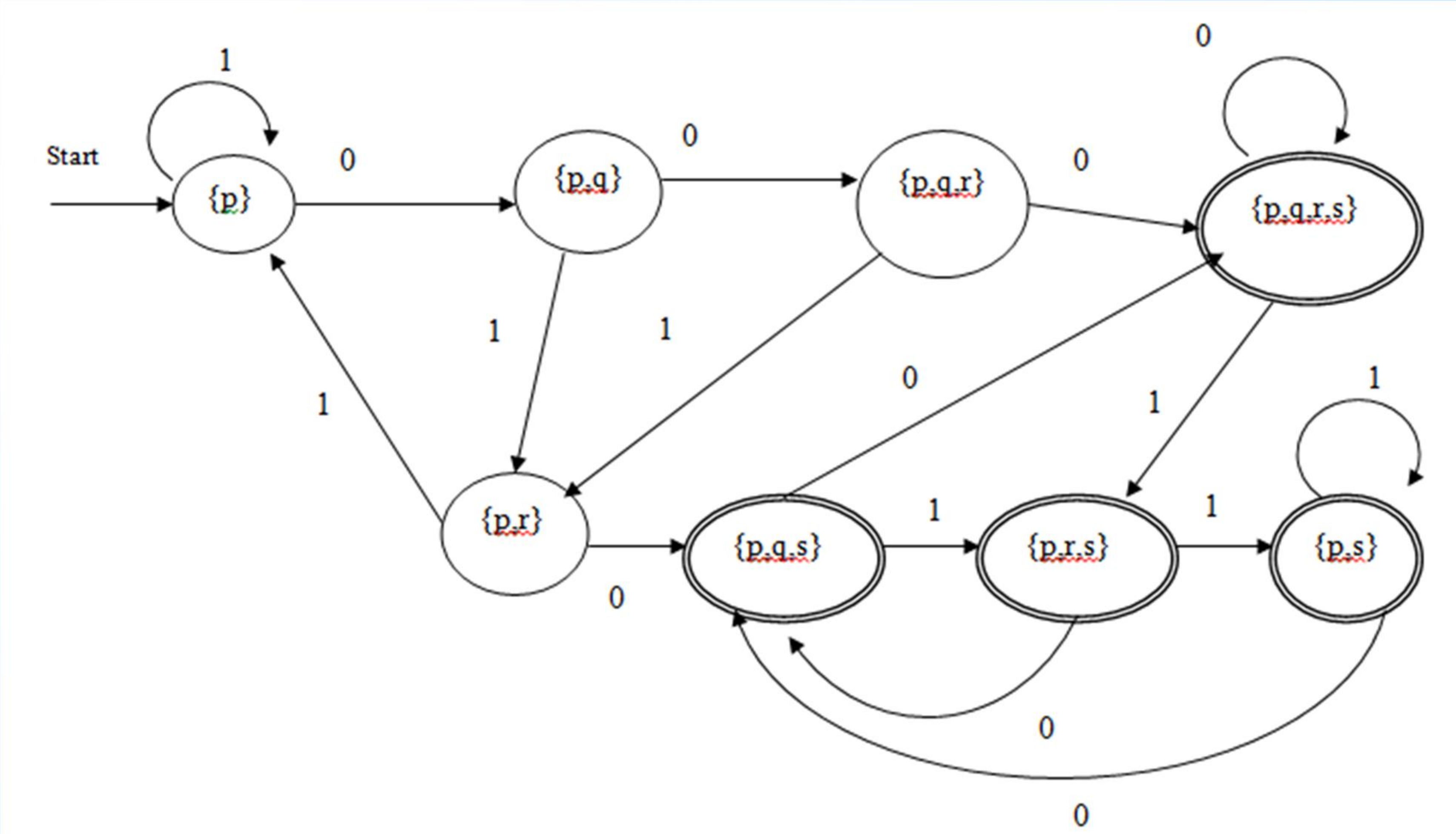
NFA

	0	1
→p	{p,q}	{p}
q	{r}	{r}
r	{s}	{}
*s	{s}	{s}

Equivalent DFA

	0	1
→ {p}	{p,q}	{p}
{p,q}	{p,q,r}	{p,r}
{p,q,r}	{p,q,r,s}	{p,r}
{p,r}	{p,q,s}	{p}
*{p,q,r,s}	{p,q,r,s}	{p,r,,s}
*{p,q,s}	{p,q,r,s}	{p,r,s}
*{p,r,s}	{p,q,s}	{p,s}
***{p,s}**	**{p,q,s}**	**{p,s}**

The transition graph of DFA - Equivalent to given NFA above



Theorem 2.11 : From Text Book Page No 63

If D=(Q_D , Σ , δ_D , $\{q_0\}$, F_D) is the DFA constructed from NFA N=**(Q_N ,** Σ , δ_N ,q_0, F_N) by the subset construction , then L(D) = L(N)

Proof: We show the fact D accepts same language as N by induction on ***/w/*** i. e. we have to show

$$\widehat{\delta_N}(q_0, w) = \widehat{\delta_D}(\{q_0\}, w)$$

Basis: w = ϵ so

$\widehat{\delta_N}(q_0, \epsilon) = \widehat{\delta_D}(\{q_0\}, \epsilon) = \{q_0\}$,

in subset construction

Proof: continued

- **Induction:** Let ***x*** is a string of length n and the relation above is true for ***x***. Let a string w such that ***w=xa*** where ***x*** is substring of w without last symbol ***'a'***. So length of ***|w|=n+1***
- Let $\boldsymbol{\widehat{\delta_N}(q_0, x) = \widehat{\delta_D}(\{q_0\}, x) = S}$ **,** where S is a subset of states in Q_N
- Let T = the union over all states p in S of $\delta_N(p, a)$.
- i.e. $\cup_{p\ in\ S}\, \delta_N(p, a)$ =T
- Then $\widehat{\delta_N}(q_0, xa) = T = \widehat{\delta_D}(\{q_0\}, w)$
  - For NFA: the extension of δ_N.
  - For DFA: definition of δ_D plus extension of δ_D.
    - That is, $\delta_D(S, a)=\delta_D(\widehat{\delta_D}\{q_0\}, x),a)= T$; then extend δ_D to w = xa.
- This completes the proof

NFA's With ϵ-Transitions

- We can allow state-to-state transitions on ϵ input.
- These transitions are done spontaneously, without looking at the input string.
- A convenience at times, but still only regular languages are accepted.

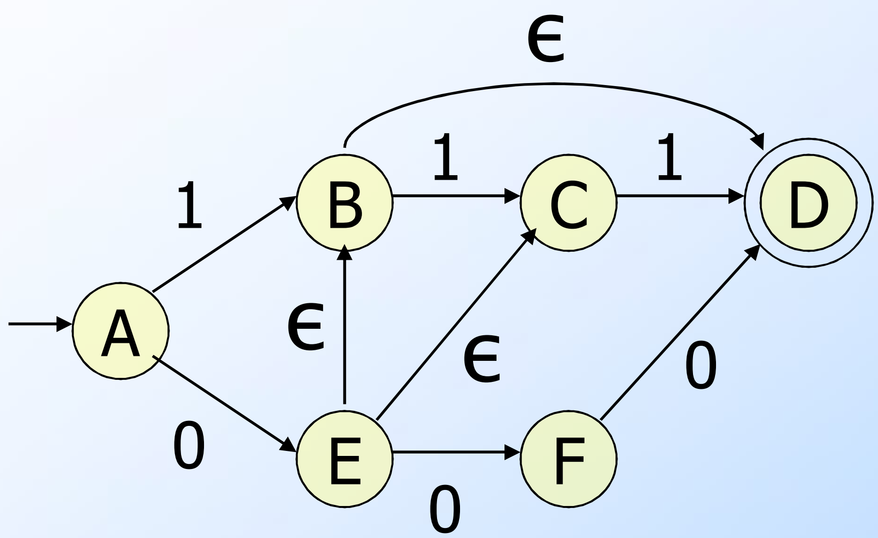
NFA with epsilon transition(ϵ-NFA)

- **Definition:** A NFA with epsilon transition(ϵ-NFA) is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where Q and ∑ are finite set of states and alphabets respectively , $q_0 \in Q$ is start state , F is set of final states and is defined as :

$$Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q$$

- The only difference between NFA and ϵ-NFA is that the transition function must include transition information on input ϵ. The ϵ, symbol for empty string , can not be a member of ∑.

Example: ϵ-NFA



		0	1	ϵ
→	A	{E}	{B}	∅
	B	∅	{C}	{D}
	C	∅	{D}	∅
*	D	∅	∅	∅
	E	{F}	∅	{B, C}
	F	{D}	∅	∅

ϵ-Closure of a state:

- The ϵ-closure of a state q is the set of all states that contains state q and can be reached by ϵ-transition along any path from q to those states obtained by one or more ϵ-transitions.
- Formally, ϵ-closure of q is defined as:

ϵ-closure(q):

- state q is in ϵ-CLOSURE(q)
- if state p is reached with -transition from state q, p is in ϵ-CLOSURE(q)
- if state p is in ϵ-CLOSURE(q) and there is a transition from state p to r labeled , then r is in ϵ-CLOSURE(q)

Closure of States: Example

- ϵ–CL(q) = set of states you can reach from state q following only arcs labeled ϵ.
- Example: ϵ–CL(A) = {A};

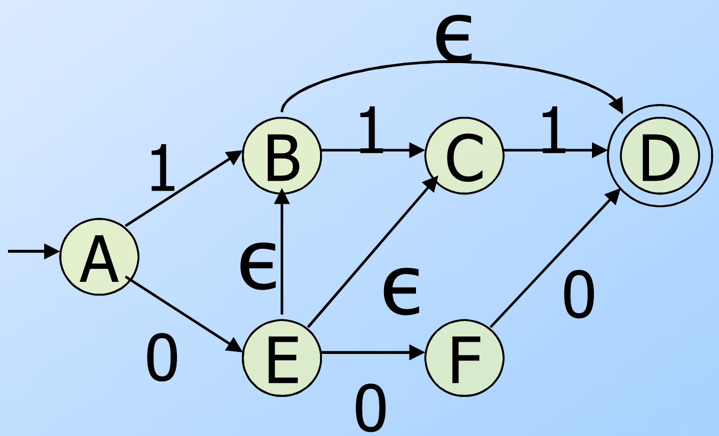
  ϵ–CL(B) = {B, D}

  ϵ–CL(C) = {C}.

  ϵ–CL(D) = {D}.

  ϵ–CL(E) = {B, C, D, E}.

  ϵ–CL(F) = {F}.



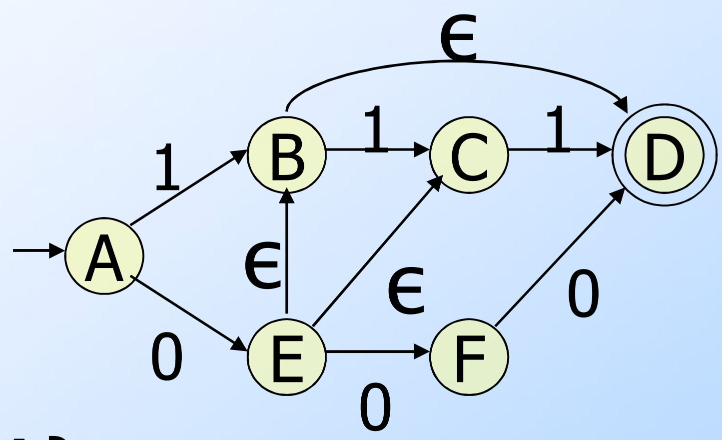
- Closure of a set of states = union of the closure of each state. E.g.

ϵ–CL({E,F}) = {B, C, D, E,F}.

Extended Delta($\hat{\delta}$)

- Basis: $\hat{\delta}(q, \epsilon) = \epsilon\text{–CL}(q)$.
- Induction: Note $a \in \Sigma$ and a not equal to empty, $\hat{\delta}(q, xa)$ is computed as follows:
  1. Start with $\hat{\delta}(q, x) = P = \{p_1, p_2, \ldots \ldots p_k\}$
  2. Take union of ϵ–CL (P) say it is $R = \{r_1, r_2, \ldots \ldots r_n\}$
  3. Take the union of $\epsilon\text{–CL}(\delta(r_i, a))$ for all r in R , let $S = \epsilon\text{–CL}(\delta(r_i, a)) = \{s_1, s_2, \ldots \ldots s_m\}$
  4. So , $\hat{\delta}(q, w) = \hat{\delta}(q, xa) = S$.
- Intuition: $\hat{\delta}(q, w)$ is the set of states you can reach from q following a path labeled w where there may be any no of ϵ within w.

Example: Extended Delta ($\hat{\delta}$)



- $\hat{\delta}$ A, ϵ) = ϵ–CL(A) = {A}.
- $\hat{\delta}$(A, 0) = ϵ–CL(δ(A,0)

  = ϵ–CL({E}) = {B, C, D, E}.
- $\hat{\delta}$ (A, 01) = ϵ–CL(δ({B,C,D,E},1)

  = ϵ–CL({C, D}) = {C, D}.
- *Language* of an ϵ-NFA is the set of strings w such that $\hat{\delta}$ (q_0, w) contains a final state.

Equivalence of NFA, ϵ-NFA

- Every NFA is an ϵ-NFA similar to "Every DFA is an NFA with exactly one transition from each state with each input.
  - i.e. It just has no transitions on ϵ.
- Converse requires us to take an ϵ-NFA and construct an NFA that accepts the same language.
- We do so by combining ϵ–transitions with the next transition on a real input i.e. removing ϵ–transitions from it.

Removing ϵ-transition from ϵ-NFA

- We can construct NFA and DFA from ϵ–NFA by removing the ϵ–transition from it.
- In NFA equivalent to given ϵ–NFA, there are as many as the states in ϵ–NFA, only ϵ–transitions are removed by using the closure of states.
- In DFA equivalent to given ϵ–NFA, it has states represented by any subset of the states of given ϵ–NFA

From ϵ-NFA to NFA

- Let $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ is an ϵ-NFA.
- To convert E into equivalent NFA,

  $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$

  do the followings.

1. Set q_0 as initial state of NFA all states are unmarked
2. for each input symbol a in Σ do

   $\delta_N(q, a) = \epsilon\text{-}CL(\delta_E(\epsilon\text{-}CL(q), a))$

1. Set $F_N = F_E$

Example: ϵ-NFA to NFA

Consider following ϵ−NFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	**ϕ**
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	**ϕ**	**ϕ**	**ϕ**

- Set A as initial state of NFA without ϵ−Transition, $N=(Q_N, \Sigma, \delta_N, q_0, F_N)$.
- So, $q_0=A$, if ϵ−NFA accepting empty string then make A also final state.
- $\delta_N(A,0)=\epsilon\text{-}CL(\delta_E(\epsilon\text{-}CL(A),0))$
  $= \epsilon\text{-}CL(\delta_E((\{A,B,D\}),0))$
  $= \epsilon\text{-}CL(\{A\} \cup \{C\} \cup \{E\})$
  $= \{A,B,D\} \cup \{C\} \cup \{E\})$
  $= \{A,B,C,D,E\}$
- $\delta_N(A,1) = \epsilon\text{-}CL(\delta_E(\epsilon\text{-}CL(A),1))$
  $= \epsilon\text{-}CL(\delta_E((\{A,B,D\}),1))$
  $= \epsilon\text{-}CL(\{E\} \cup \{D\})$
  $= \{E\} \cup \{D\}$
  $= \{D,E\}$

Example: ϵ-NFA to NFA

- **Consider following** ϵ−NFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	ϕ	ϕ	ϕ

- $\delta_N(B,0) = \epsilon\text{-CL}(\delta_E(\epsilon\text{-CL}(B),0))$

$$= \epsilon\text{-CL}(\delta_E(\{B\},0))$$
$$= \epsilon\text{-CL}(\{C\})$$
$$= \{C\}$$

- $\delta_N(B,1) = \epsilon\text{-CL}(\delta_E(\epsilon\text{-CL}(B),1))$

$$= \epsilon\text{-CL}(\delta_E(\{B\},1))$$
$$= \epsilon\text{-CL}(\{E\})$$
$$= \{E\}$$

Example: ϵ-**NFA to NFA**

- **Consider following** ϵ−NFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	ϕ	ϕ	ϕ

- $\delta_N(C,0) = \epsilon\text{-CL}(\delta_E(\epsilon\text{-CL}(C),0))$
  $= \epsilon\text{-CL}(\delta_E(\{C\},0))$
  $= \epsilon\text{-CL}(\phi)$
  $= \phi$

- $\delta_N(C,1) = \epsilon\text{-CL}(\delta_E(\epsilon\text{-CL}(C),1))$
  $= \epsilon\text{-CL}(\delta_E(\{C\},1))$
  $= \epsilon\text{-CL}(\{B\})$
  $= \{B\}$

Example: ϵ-NFA to NFA

- **Consider following** ϵ–NFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	ϕ	ϕ	ϕ

- $\delta_N(D,0)=\epsilon\text{-CL}(\delta_E(\epsilon\text{-CL}(D),0))$

$$= \epsilon\text{-CL}(\delta_E(\{D\},0))$$

$$= \epsilon\text{-CL}(\{E\})$$

$$= \{E\}$$

- $\delta_N(D,1)=\epsilon\text{-CL}(\delta_E(\epsilon\text{-CL}(D),1))$

$$= \epsilon\text{-CL}(\delta_E(\{D\},1))$$

$$= \epsilon\text{-CL}(\{D\})$$

$$= \{D\}$$

Example: ϵ-NFA to NFA

- **Consider following** ϵ−NFA

Q	ϵ	0	1
→A	{B,D}	{A}	ϕ
B	ϕ	{C}	{E}
C	ϕ	ϕ	{B}
D	ϕ	{E}	{D}
*E	ϕ	ϕ	ϕ

- $\delta_N(E,0) = \epsilon\text{-CL}(\delta_E(\epsilon\text{-CL}(E),0))$
  $= \epsilon\text{-CL}(\delta_E(\{E\},0))$
  $= \epsilon\text{-CL}(\phi)$
  $= \phi$

- $\delta_N(E,1) = \epsilon\text{-CL}(\delta_E(\epsilon\text{-CL}(E),1))$
  $= \epsilon\text{-CL}(\delta_E(\{E\},1))$
  $= \epsilon\text{-CL}(\phi)$
  $= \phi$

Example: ϵ-NFA to NFA

- **Consider following** ϵ–NFA

Q	ϵ	0	1
→A	{B,D}	{A}	ϕ
B	ϕ	{C}	{E}
C	ϕ	ϕ	{B}
D	ϕ	{E}	{D}
*E	ϕ	ϕ	ϕ

Hence We got following transitions for NFA.

- $\delta_N(A,0)=\{A,B,C,D,E\}$
- $\delta_N(A,1)=\{D,E\}$
- $\delta_N(B,0)=\{C\}$
- $\delta_N(B,1)=\{E\}$
- $\delta_N(C,0)=\phi$
- $\delta_N(C,1)=\{B\}$
- $\delta_N(D,0)=\{E\}$
- $\delta_N(D,1)=\{D\}$
- $\delta_N(E,0)=\phi$
- $\delta_N(E,1)=\phi$

Example: ϵ-NFA to NFA

- **Consider following** ϵ–NFA

Hence We got following transitions for NFA.

- $\delta_N(A,0)=\{A,B,C,D,E\}$
- $\delta_N(A,1)=\{D,E\}$
- $\delta_N(B,0)=\{C\}$
- $\delta_N(B,1)=\{E\}$
- $\delta_N(C,0)=\phi$
- $\delta_N(C,1)=\{B\}$
- $\delta_N(D,0)=\{E\}$
- $\delta_N(D,1)=\{D\}$
- $\delta_N(E,0)=\phi$
- $\delta_N(E,1)=\phi$

Q	**0**	**1**
→A	{A,B,C,D,E}	**{D,E}**
B	{C}	**{E}**
C	ϕ	**{B}**
D	{E}	**{D}**
***E**	ϕ	ϕ

Table NFA equivalent to given ϵ–NFA

From ϵ–NFA to DFA

- **<u>Method 1:</u>**
  1. Convert ϵ–NFA to NFA as described above.
  2. Convert the NFA obtained in 1 to DFA using Subset Construction as described in Subset Construction method
- <u>Method 2:</u>

1. Convert ϵ–NFA direct to DFA as described in next slide.

From ϵ-NFA to DFA

- Let $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ is an ϵ-NFA.
- To convert E directly into equivalent DFA,

  $D = (Q_D, \Sigma, \delta_D, q_s, F_N)$

  do the followings.

1. Set $q_s = \epsilon\text{-}CL(q_0)$ as initial state of DFA
2. for each input symbol a in Σ, each state in DFA $S = \{p_1, p_2, \ldots, p_k\}$ do

   $\delta_D(S, a) = \epsilon\text{-}CL(\delta_E(S, a))$

   i.e. $\delta_D(S, a) = \epsilon\text{-}CL(\cup_{i=1}^{k} \delta_E(p_i, a))$

3. Set F_D = those states of Q_D that contains at least one accepting state of E i.e. $F_D = \{S \mid S \text{ is in } Q_D \text{ and } S \cap F_E \neq \phi\}$

From ϵ-NFA to DFA: Example

- **Consider the following** ϵ–NFA, we have to convert it into DFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	**ϕ**	**ϕ**	**ϕ**

Start state in D =ϵ-CL(A)={A,B,D}

States/Inp	**0**	**1**
→{A,B,D}		

From ϵ-NFA to DFA: Example

- **Consider the following** ϵ–NFA, we have to convert it into DFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	ϕ	ϕ	ϕ

States/Inp	0	1
→{A,B,D}	{A,B,C,D,E}	{D,E}
*{A,B,C,D,E}		
*{D,E}		

From ϵ-NFA to DFA: Example

- **Consider the following** ϵ–NFA, we have to convert it into DFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	**ϕ**	**ϕ**	**ϕ**

States/Inp	0	1
→{A,B,D}	{A,B,C,D,E}	{D,E}
***{A,B,C,D,E}**	**{A,B,C,D,E}**	**{B,D,E}**
*{D,E}		
***{B,D,E}**		

From ϵ-NFA to DFA: Example

- **Consider the following** ϵ–NFA, we have to convert it into DFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	**ϕ**	**ϕ**	**ϕ**

States/Inp	0	1
→{A,B,D}	{A,B,C,D,E}	{D,E}
*{A,B,C,D,E}	{A,B,C,D,E}	{B,D,E}
***{D,E}**	**{E}**	**{D}**
*{B,D,E}		
***{E}**		
{D}		

From ϵ-NFA to DFA: Example

- **Consider the following** ϵ–NFA, we have to convert it into DFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	ϕ	ϕ	ϕ

States/Inp	**0**	**1**
→{A,B,D}	{A,B,C,D,E}	{D,E}
*{A,B,C,D,E}	{A,B,C,D,E}	{B,D,E}
*{D,E}	{E}	{D}
***{B,D,E}**	**{C,E}**	**{D,E}**
*{E}		
{D}		
***{C,E}**		

From ϵ-NFA to DFA: Example

- **Consider the following** ϵ–NFA, we have to convert it into DFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	ϕ	ϕ	ϕ

States/Inp	0	1
→{A,B,D}	{A,B,C,D,E}	{D,E}
*{A,B,C,D,E}	{A,B,C,D,E}	{B,D,E}
*{D,E}	{E}	{D}
*{B,D,E}	{C,E}	{D,E}
***{E}**	ϕ	ϕ
{D}		
*{C,E}		
ϕ		

From ϵ-NFA to DFA: Example

- **Consider the following** ϵ–NFA, we have to convert it into DFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	ϕ	ϕ	ϕ

States/Inp	0	1
→{A,B,D}	{A,B,C,D,E}	{D,E}
*{A,B,C,D,E}	{A,B,C,D,E}	{B,D,E}
*{D,E}	{E}	{D}
*{B,D,E}	{C,E}	{D,E}
*{E}	ϕ	ϕ
{D}	**{E}**	**{D}**
*{C,E}		
ϕ		

From ϵ-NFA to DFA: Example

- **Consider the following** ϵ–NFA, we have to convert it into DFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	ϕ	ϕ	ϕ

States/Inp	0	1
→{A,B,D}	{A,B,C,D,E}	{D,E}
*{A,B,C,D,E}	{A,B,C,D,E}	{B,D,E}
*{D,E}	{E}	{D}
*{B,D,E}	{C,E}	{D,E}
*{E}	ϕ	ϕ
{D}	{E}	{D}
***{C,E}**	ϕ	**{B}**
ϕ		
{B}		

From ϵ-NFA to DFA: Example

- **Consider the following** ϵ–NFA, we have to convert it into DFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	ϕ	ϕ	ϕ

States/Inp	0	1
→{A,B,D}	{A,B,C,D,E}	{D,E}
*{A,B,C,D,E}	{A,B,C,D,E}	{B,D,E}
*{D,E}	{E}	{D}
*{B,D,E}	{C,E}	{D,E}
*{E}	ϕ	ϕ
{D}	{E}	{D}
*{C,E}	ϕ	{B}
ϕ	ϕ	ϕ
{B}		

From ϵ-NFA to DFA: Example

- **Consider the following** ϵ–NFA, we have to convert it into DFA

Q	ϵ	0	1
→**A**	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
***E**	ϕ	ϕ	ϕ

States/Inp	**0**	**1**
→{A,B,D}	{A,B,C,D,E}	{D,E}
*{A,B,C,D,E}	{A,B,C,D,E}	{B,D,E}
*{D,E}	{E}	{D}
*{B,D,E}	{C,E}	{D,E}
*{E}	ϕ	ϕ
{D}	{E}	{D}
*{C,E}	ϕ	{B}
ϕ	ϕ	ϕ
{B}	{C}	**{E}**
{C}		

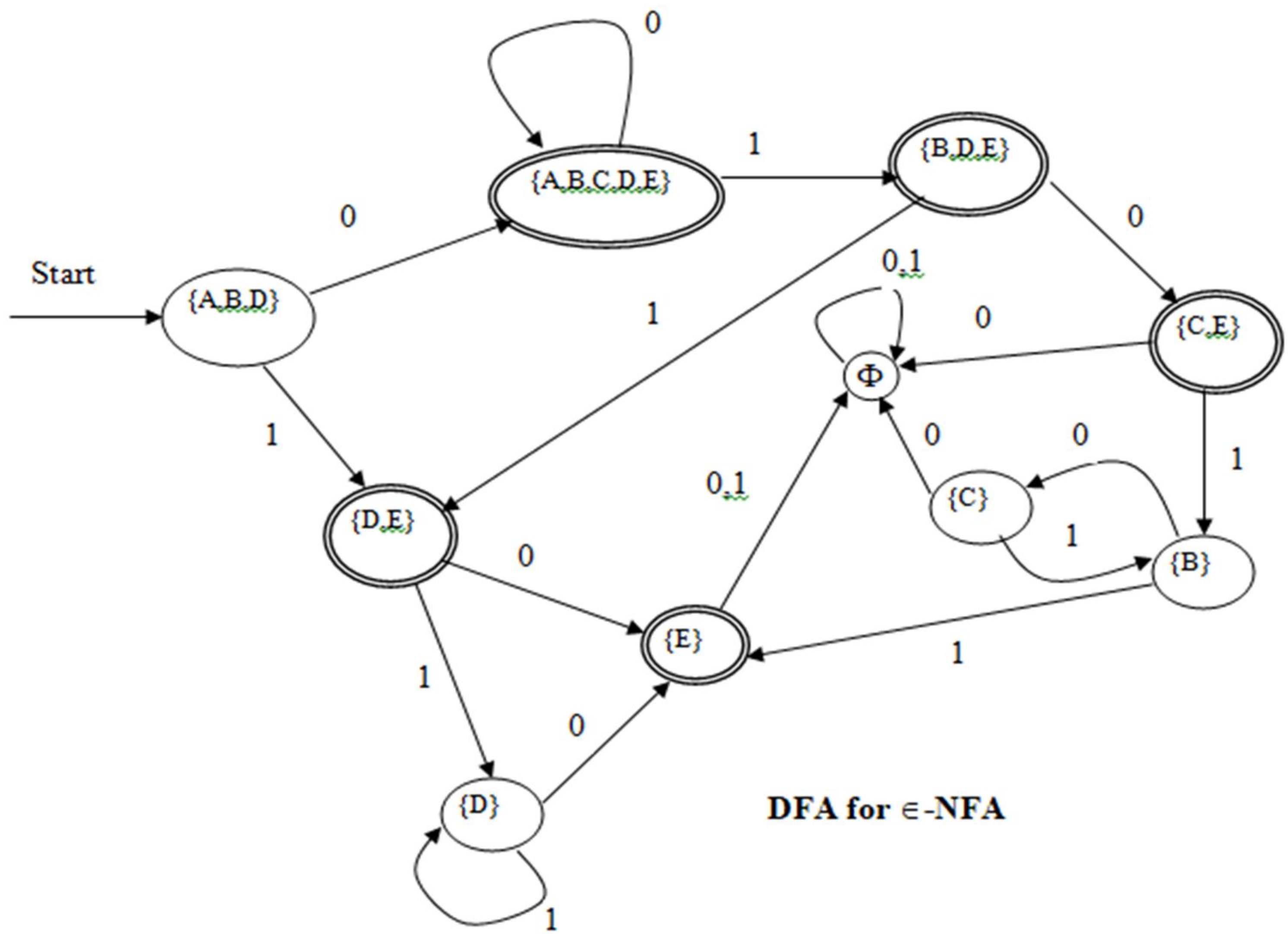
From ϵ-NFA to DFA: Example

- **Consider the following** ϵ–NFA, we have to convert it into DFA

Q	ϵ	0	1
→A	{B,D}	{A}	ϕ
B	ϕ	{C}	**{E}**
C	ϕ	ϕ	**{B}**
D	ϕ	{E}	**{D}**
*E	ϕ	ϕ	ϕ

States/Inp	0	1
→{A,B,D}	{A,B,C,D,E}	{D,E}
*{A,B,C,D,E}	{A,B,C,D,E}	{B,D,E}
*{D,E}	{E}	{D}
*{B,D,E}	{C,E}	{D,E}
*{E}	ϕ	ϕ
{D}	{E}	{D}
*{C,E}	ϕ	{B}
ϕ	ϕ	ϕ
{B}	{C}	{E}
{C}	**ϕ**	**{B}**

Transition Diagram of DFA constructed from ϵ–NFA



DFA for ∈-NFA

Summary

- DFA's, NFA's, and ϵ–NFA's all accept exactly the same set of languages: the regular languages.
- The NFA types are easier to design and may have exponentially fewer states than a DFA.
- ϵ–NFA are most flexible model of FA to design and construct for any regular language.
- The computation power of all three models are same NFA and ϵ–NFA only adds the flexibility to construct FA for a language.