Finite Automata

Deterministic Finite Automata (DFA)

- A Deterministic Finite Automata (DFA) is a quintuple

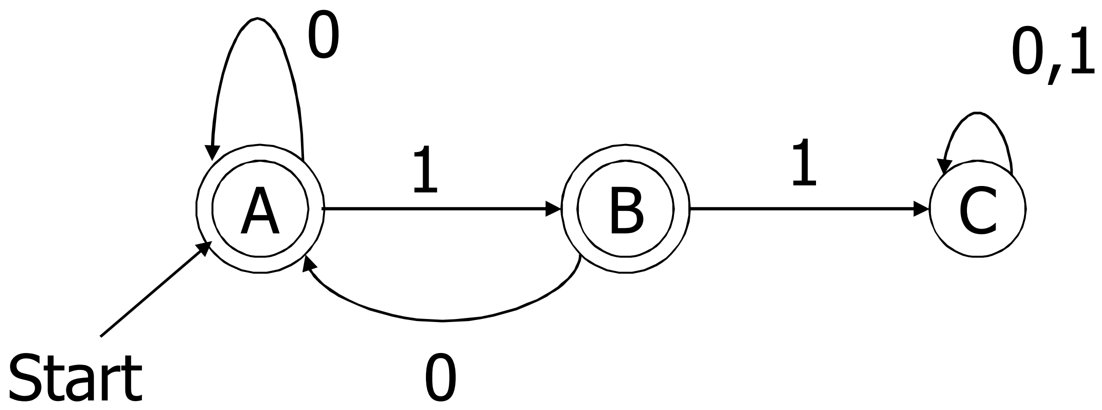
  $$\mathbf{A = (Q, \Sigma, \delta, q_0, F)}$$

1. Q is a finite set of states
2. Σ is a finite set of symbols (alphabet)
3. Delta (δ) is a transition function $(q,a) \rightarrow p$
4. q_0 is the start state ($q_0 \in Q$)
5. F is a set of final (accepting) states ($F \subseteq Q$)

- Transition function takes two arguments: a state and an input symbol.
- $\delta(q, a)$ = the state that the DFA goes to when it is in state q and input a is received.

Graph Representation of DFA's

- Nodes = states.
- Arcs represent transition function.
- Arc from state p to state q labeled by all those input symbols that have transitions from p to q.
- Arrow labeled "Start" to the start state.
- Final states indicated by double circles.



Alternative Representation: Transition Table

Final states starred

Columns = input symbols

Arrow for start state

	0	1
→ * A	A	B
* B	A	C
C	C	C

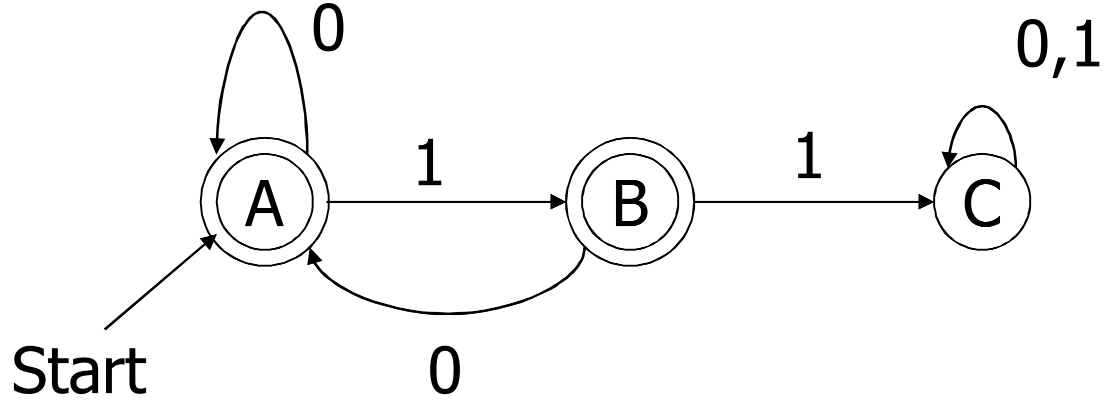
Rows = states

Strings Accepted By A DFA

- An DFA accepts a string $w = a_1a_2 \ldots a_n$ if its path in the transition diagram that
  1. Begins at the start state
  2. Ends at an accepting state

- This DFA accepts
  input: 010001

- This DFA rejects
  input: 011001



Extended Delta Function – Delta Hat

- The transition function δ can be extended to $\hat{\delta}$ that operates on states and strings (as opposed to states and symbols)

$$\hat{\delta}(q,\varepsilon) = q$$

$$\hat{\delta}(q,xa) = \delta(\hat{\delta}(q,x), a)$$

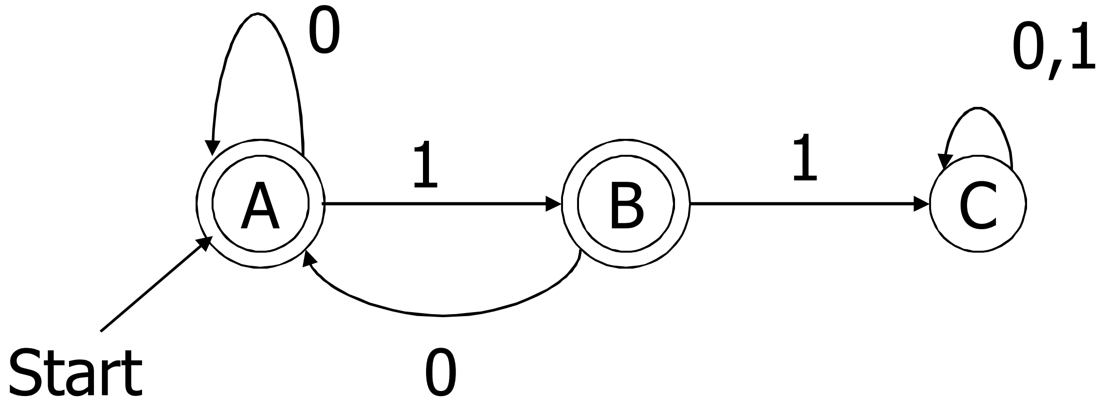
Language Accepted by a DFA

- Formally, the language accepted by a DFA A is

$$L(A) = \{ w \mid \hat{\delta}(q_0,w) \in F \}$$

- Languages accepted by DFAs are called as **regular languages**.
  - Every DFA accepts a regular language, and
  - For every regular language there is a DFA accepts it

Language Accepted by a DFA

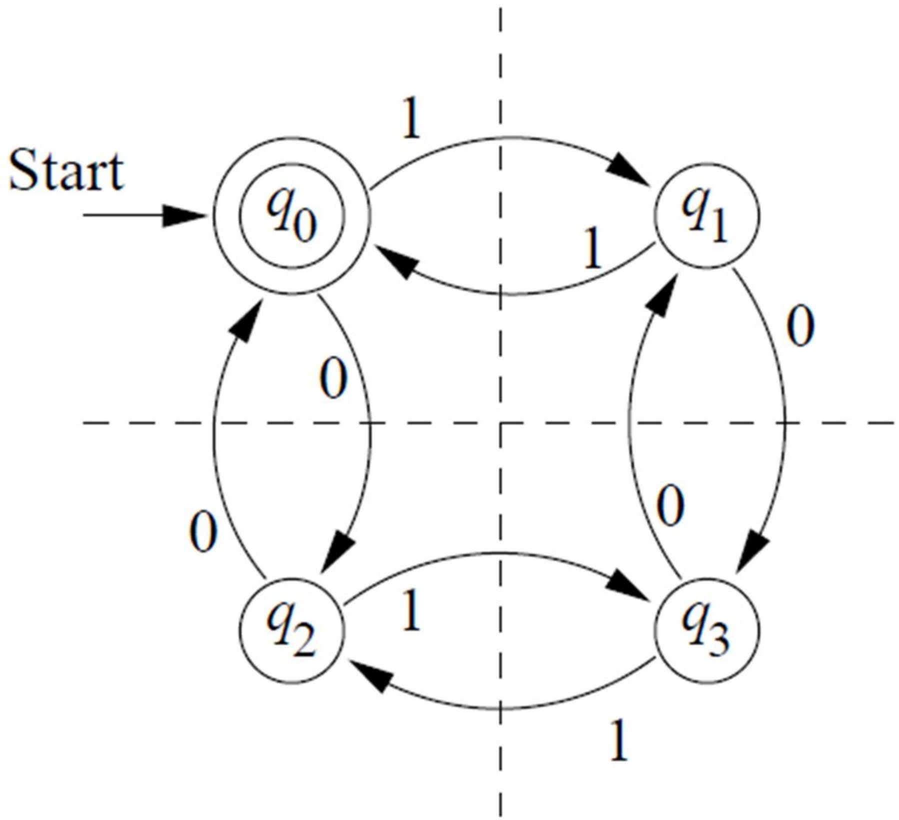


- This DFA accepts all strings of 0's and 1's without two consecutive 1's.
- Formally,

  L(A) = { w |w is in {0,1}* and w does not have two consecutive 1's }

DFA Example

- A DFA accepting all and only strings with an even number of 0's and an even number of 1's



Tabular representation of the DFA

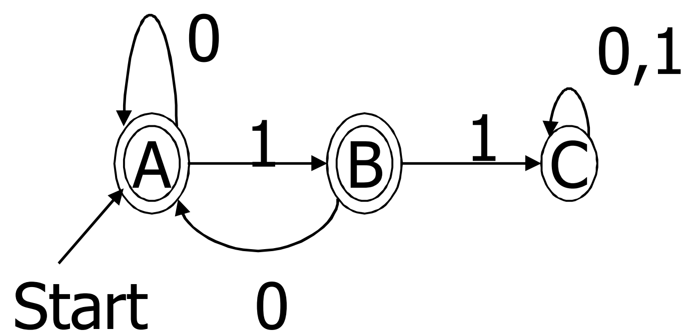
	0	1
$\star \rightarrow q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

Proofs of Set Equivalence

- We need to prove that two descriptions of sets are in fact the same set.
- Here, one set is "the language of this DFA," and the other is "the set of strings of 0's and 1's with no consecutive 1's."
- In general, to prove S=T, we need to prove two parts:

  $S \subseteq T$ and $T \subseteq S$. That is:

  1. If w is in S, then w is in T.
  2. If w is in T, then w is in S.
- As an example, let S = the language of our running DFA, and T = "no consecutive 1's."



Proof Part 1 : $S \subseteq T$

- **To prove**: if w is accepted by our DFA
  then w has no consecutive 1's.

- Proof is an induction on length of w.

- **Important trick**: Expand the inductive hypothesis to be more detailed than you need.

The Inductive Hypothesis

1. If $\hat{\delta}(A, w) = A$, then w has no consecutive 1's and does not end in 1.
2. If $\hat{\delta}(A, w) = B$, then w has no consecutive 1's and ends in a single 1.

- **Basis**: $|w| = 0$; i.e., $w = \epsilon$.
  - (1) holds since ϵ has no 1's at all.
  - (2) holds *vacuously*, since $\hat{\delta}(A, \epsilon)$ is not B.

Important concept:
If the "if" part of "if..then" is false,
the statement is true.

Inductive Step

- Need to prove (1) and (2) for w = xa.

- (1) for w is: If $\hat{\delta}(A, w) = A$, then w has no consecutive 1's and does not end in 1.

- Since $\hat{\delta}(A, w) = A$, $\hat{\delta}(A, x)$ must be A or B, and a must be 0 (look at the DFA).

- By the IH, x has no 11's.

- Thus, w has no 11's and does not end in 1.

Inductive Step

- Now, prove (2) for w = xa: If $\hat{\delta}$(A, w) = B, then w has no 11's and ends in 1.

- Since $\hat{\delta}$(A, w) = B, $\hat{\delta}$(A, x) must be A, and *a* must be 1 (look at the DFA).

- By the IH, x has no 11's and does not end in 1.

- Thus, w has no 11's and ends in 1.

Proof Part 1 : $T \subseteq S$

- Now, we must prove:

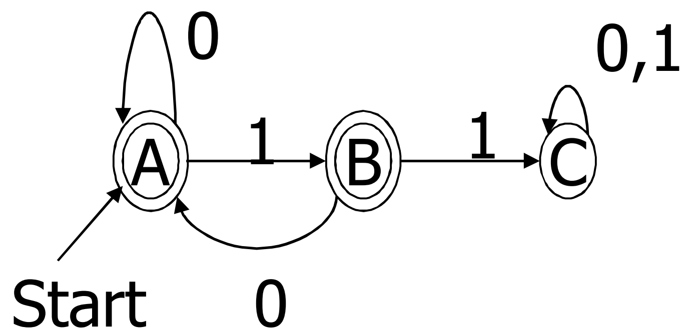
  if w has **no** 11's, then w is accepted by our DFA

- *Contrapositive* : If w is **not** accepted by our DFA then w has 11.

Key idea: contrapositive
of "if X then Y" is the
equivalent statement
"if not Y then not X."

Using the Contrapositive

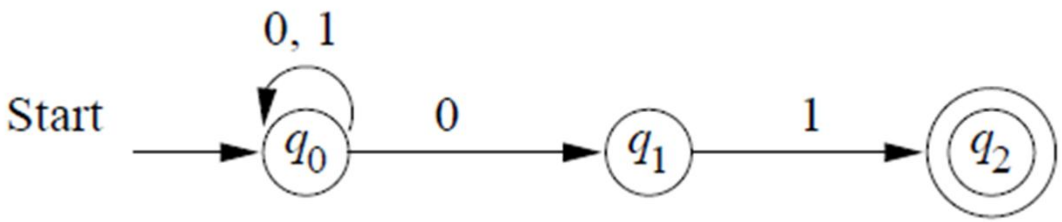
- Every w gets the DFA to exactly one state.
  - Simple inductive proof based on:
    - Every state has exactly one transition on 1, one transition on 0.
- The only way w is not accepted is if it gets to C.
- The only way to get to C [formally: $\hat{\delta}(A,w) = C$] is if w = x1y, x gets to B, and y is the tail of w that follows what gets to C for the first time.
- If $\hat{\delta}(A,x) = B$ then surely x = z1 for some z.
- Thus, w = z11y and has 11.



Nondeterministic Finite Automata (NFA)

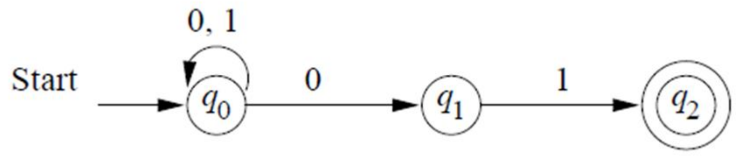
- A NFA can be in several states at once, or, it can "guess" which state to go to next.
- A NFA state can have more than one arc leaving from that state with a same symbol.

- *Example*: An automaton that accepts all and only strings ending in 01.

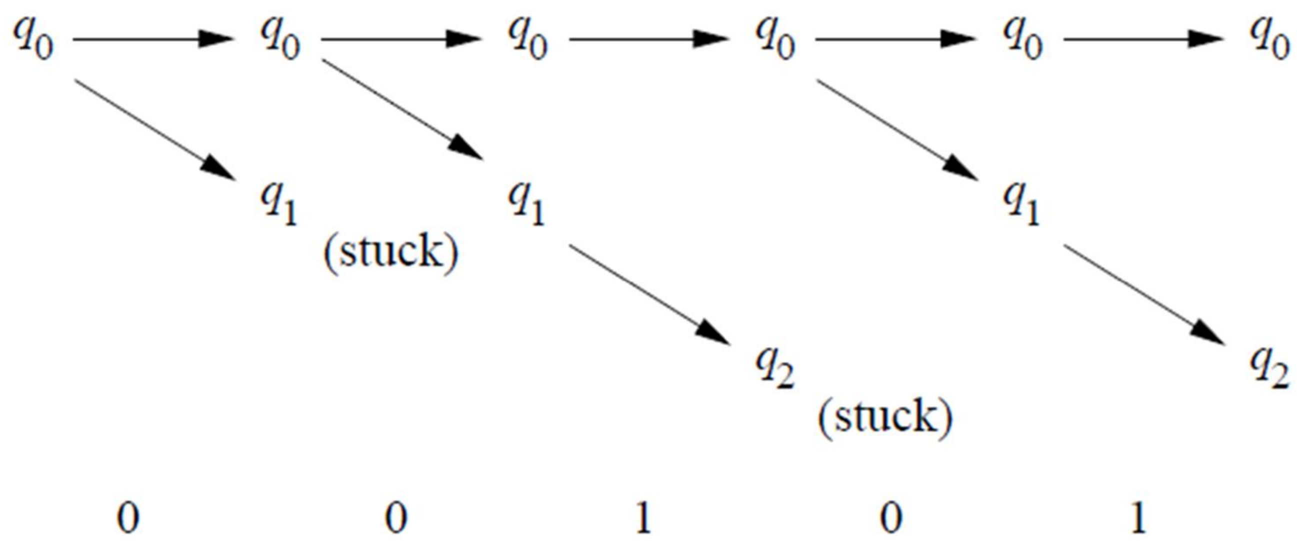


- State q_0 can go to q_0 or q_1 with the symbol 0.

NFA – Example



- What happens when the NFA processes the input 00101



- In fact, all missing arcs go to a death state, the death state goes to itself for all symbols, and the death state is a non-accepting state.

Definition of Nondeterministic Finite Automata

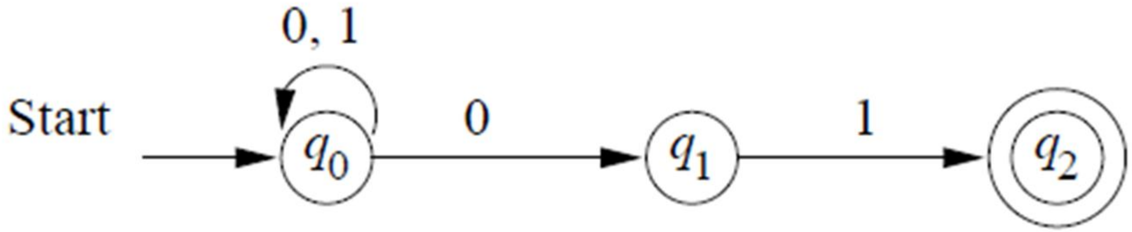
- Formally, a Nondeterministic Finite Automata (NFA) is a quintuple

  $$\mathbf{A = (Q, \Sigma, \delta, q_0, F)}$$

1. Q is a finite set of states
2. Σ is a finite set of symbols (alphabet)
3. Delta (δ) is a transition function from $Q x \Sigma$ to the powerset of Q.
4. q_0 is the start state ($q_0 \in Q$)
5. F is a set of final (accepting) states ($F \subseteq Q$)

- Transition function takes two arguments: a state and an input symbol.
- $\delta(q, a)$ = the set of the states that the DFA goes to when it is in state *q* and input *a* is received.

NFA – Table Representation



The table representation of this NFA is as follows.

NFA is $(\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$

its transition function is

	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
$\star q_2$	$\emptyset$	$\emptyset$

Extended Transition Function for NFA – Delta Hat

- The transition function δ can be extended to $\hat{\delta}$ that operates on states and strings (as opposed to states and symbols)

Basis: $\hat{\delta}(q,\varepsilon) = q$

Induction: If $\hat{\delta}(q,x) = \{p_1,p_2,...,p_k\}$ for a string x, then

$$\hat{\delta}(q,xa) = \bigcup_{i=1}^{k} \delta(p_i, a)$$

For the string w = xa, we compute $\hat{\delta}(q,x)$ first, then we follow any transition from any of the states with the symbol a.

Language of a NFA

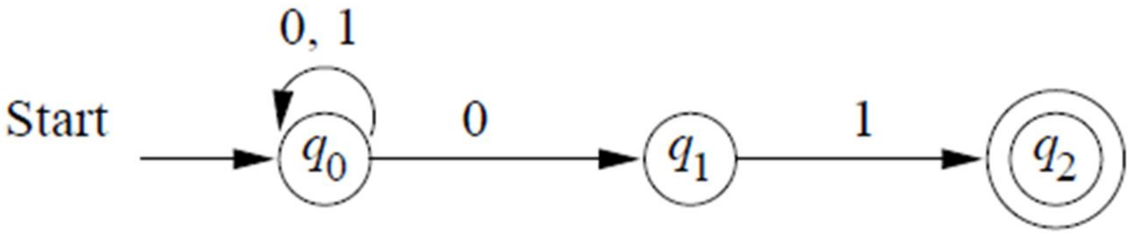
- The language accepted by a NFA A is

$$L(A) = \{w : \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$$

- i.e. a string w is accepted by a NFA A iff the states that are reachable from the starting state by consuming w contain at least one final state.

Language of a NFA - Example

- Let's prove formally that the NFA



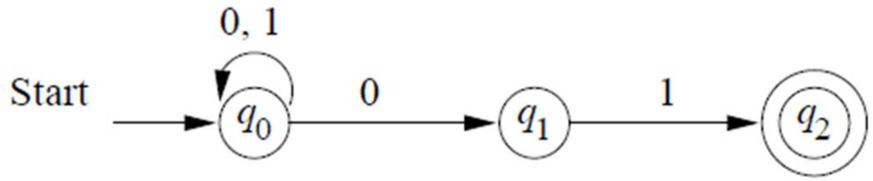
accepts the language $\{x01 : x \in \Sigma^* \}$. We'll do a mutual induction on the following three statements,

$$w \in \Sigma^* \Rightarrow q_0 \in \hat{\delta}(q_0, w)$$

$$q_1 \in \hat{\delta}(q_0, w) \Leftrightarrow w = x0$$

$$q_2 \in \hat{\delta}(q_0, w) \Leftrightarrow w = x01$$

Proof



BASIS: If $|w| = 0$, then $w = \epsilon$. Statement (1) says that $\hat{\delta}(q_0, \epsilon)$ contains q_0, which it does by the basis part of the definition of $\hat{\delta}$. For statement (2), we know that ϵ does not end in 0, and we also know that $\hat{\delta}(q_0, \epsilon)$ does not contain q_1, again by the basis part of the definition of $\hat{\delta}$. Thus, the hypotheses of both directions of the if-and-only-if statement are false, and therefore both directions of the statement are true. The proof of statement (3) for $w = \epsilon$ is essentially the same as the above proof for statement (2).

INDUCTION: Assume that $w = xa$, where a is a symbol, either 0 or 1. We may assume statements (1) through (3) hold for x, and we need to prove them for w. That is, we assume $|w| = n + 1$, so $|x| = n$. We assume the inductive hypothesis for n and prove it for $n + 1$.

1. We know that $\hat{\delta}(q_0, x)$ contains q_0. Since there are transitions on both 0 and 1 from q_0 to itself, it follows that $\hat{\delta}(q_0, w)$ also contains q_0, so statement (1) is proved for w.

2. (If) Assume that w ends in 0; i.e., $a = 0$. By statement (1) applied to x, we know that $\hat{\delta}(q_0, x)$ contains q_0. Since there is a transition from q_0 to q_1 on input 0, we conclude that $\hat{\delta}(q_0, w)$ contains q_1.

   (Only-if) Suppose $\hat{\delta}(q_0, w)$ contains q_1. If we look at the diagram of Fig. 2.9, we see that the only way to get into state q_1 is if the input sequence w is of the form $x0$. That is enough to prove the "only-if" portion of statement (2).

3. (If) Assume that w ends in 01. Then if $w = xa$, we know that $a = 1$ and x ends in 0. By statement (2) applied to x, we know that $\hat{\delta}(q_0, x)$ contains q_1. Since there is a transition from q_1 to q_2 on input 1, we conclude that $\hat{\delta}(q_0, w)$ contains q_2.

   (Only-if) Suppose $\hat{\delta}(q_0, w)$ contains q_2. Looking at the diagram of Fig. 2.9, we discover that the only way to get to state q_2 is for w to be of the form $x1$, where $\hat{\delta}(q_0, x)$ contains q_1. By statement (2) applied to x, we know that x ends in 0. Thus, w ends in 01, and we have proved statement (3).

Equivalence of DFA and NFA

- NFA's are usually easier to construct.
- Surprisingly, for any NFA N there is a DFA D, such that L(D) = L(N), and vice versa.
- This involves the subset construction, an important example how an automaton B can be generically constructed from another automaton A.
- Given an NFA

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

we will constract a DFA

$$D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$$

such that L(D) = L(N)

Subset Construction

- $Q_D = \{S : S \subseteq Q_N\}$.

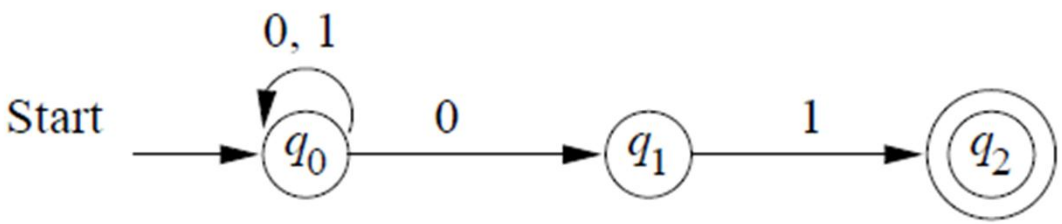
Note: $|Q_D| = 2^{|Q_N|}$, although most states in Q_D are likely to be garbage.

- $F_D = \{S \subseteq Q_N : S \cap F_N \neq \emptyset\}$

- For every $S \subseteq Q_N$ and $a \in \Sigma$,

$$\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$$

Subset Construction - Example



	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_2\}$
$\star\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\star\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\star\{q_1, q_2\}$	$\emptyset$	$\{q_2\}$
$\star\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

Subset Construction – Accessible States

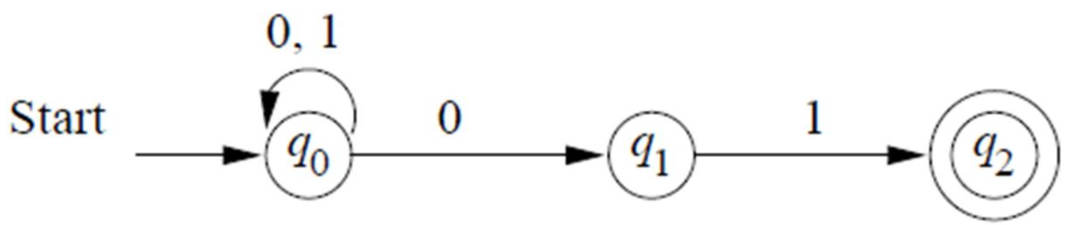
- We can often avoid the exponential blow-up by constructing the transition table for D only for accessible states S as follows:

Basis: $S = \{q_0\}$ is accessible in D

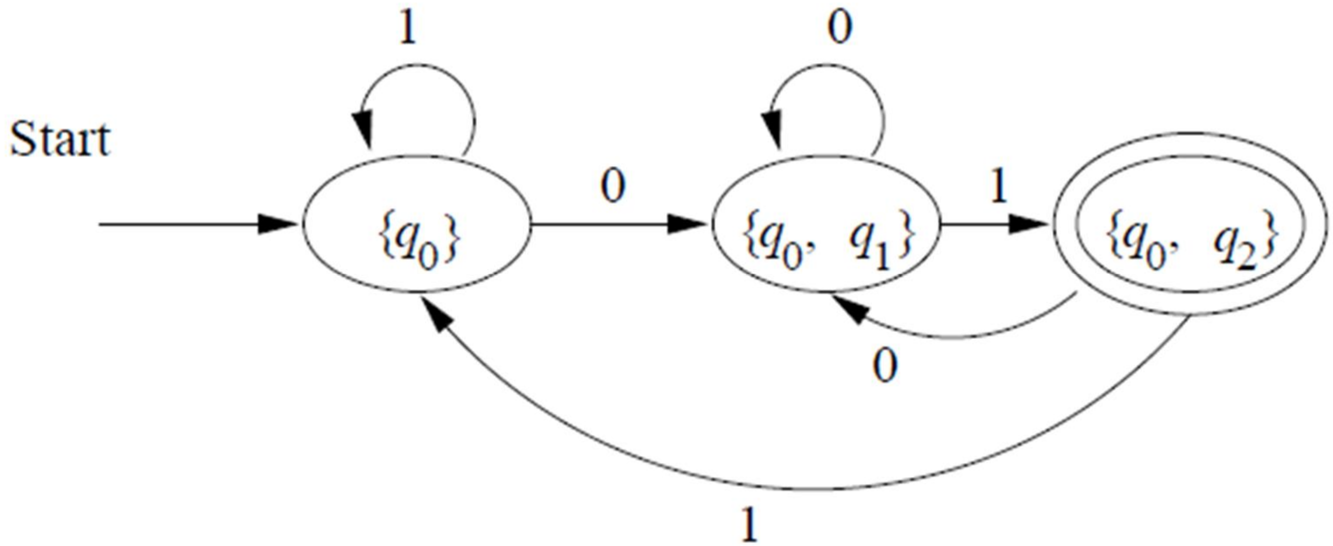
Induction: If state S is accessible, so are the states in $\bigcup_{a \in \Sigma} \delta_D(S, a)$.

Subset Construction – Accessible States (example)

NFA



DFA



Equivalence of DFA and NFA - Theorem

Theorem: Let D be the *subset* DFA of an NFA N. Then L(D) = L(N).

Proof: We show on an induction on |w| that

$$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$$

Basis: w = ε. The claim follows from definition.

Induction:

$$\begin{aligned}
\hat{\delta}_D(\{q_0\}, xa) &\overset{\text{def}}{=} \delta_D(\hat{\delta}_D(\{q_0\}, x), a) \\
&\overset{\text{i.h.}}{=} \delta_D(\hat{\delta}_N(q_0, x), a) \\
&\overset{\text{cst}}{=} \bigcup_{p \in \hat{\delta}_N(q_0, x)} \delta_N(p, a) \\
&\overset{\text{def}}{=} \hat{\delta}_N(q_0, xa)
\end{aligned}$$

So, L(D) =L(N)

Equivalence of DFA and NFA – Theorem 2

Theorem: A language L is accepted by some DFA if and only if L is accepted by some NFA.

Proof: The if-part is proved by the previous theorem.

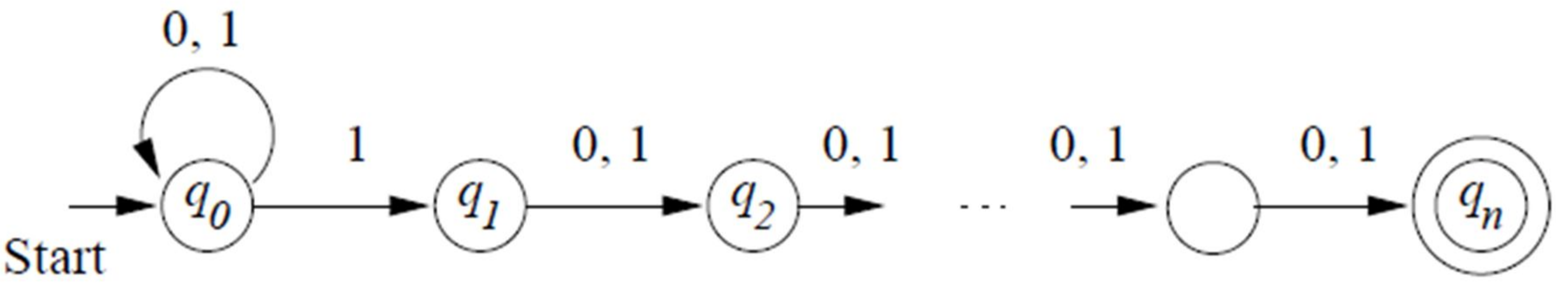
For the only-if-part, we note that any DFA can be converted to an equivalent NFA by modifying the δ_D to δ_N by the rule

If $\delta_D(q, a) = p$, then $\delta_N(q, a) = \{p\}$.

By induction on $|w|$ it will be shown in the tutorial that if $\hat{\delta}_D(q_0, w) = p$, then $\hat{\delta}_N(q_0, w) = \{p\}$.

A Bad Case for Subset Construction - Exponential Blow-Up

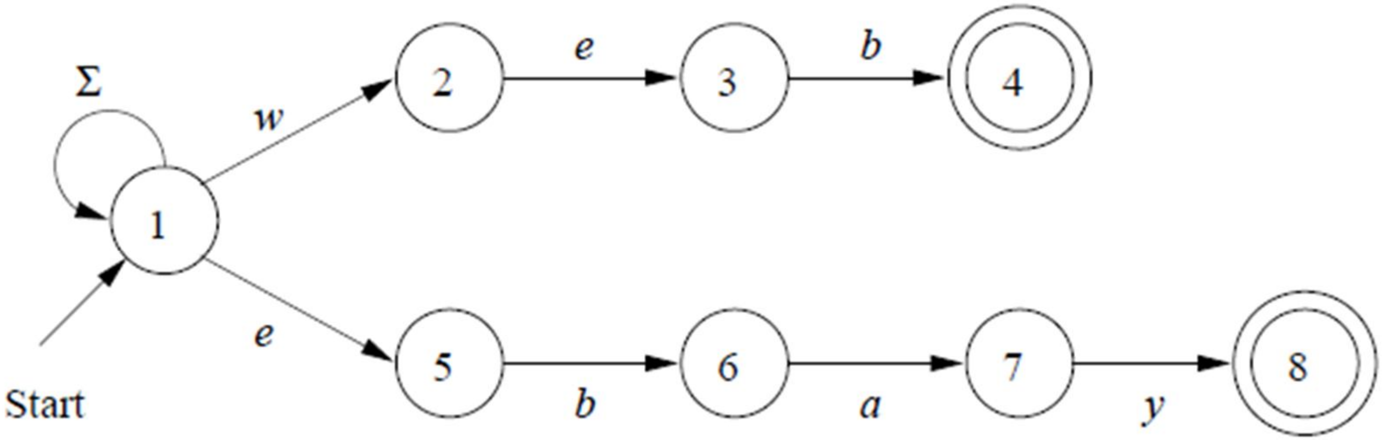
- There is an NFA N with n+1 states that has no equivalent DFA with fewer than 2^n states



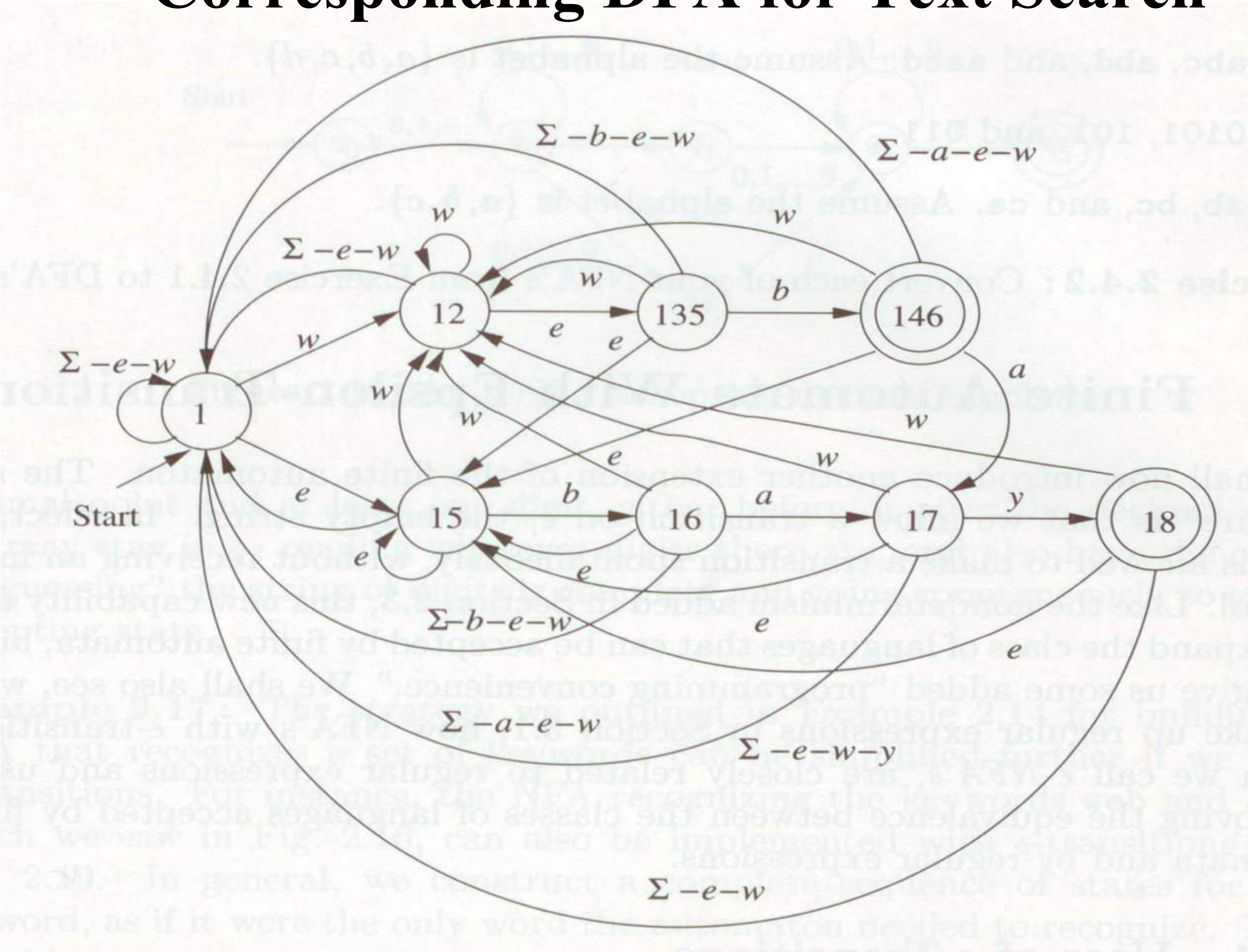
$$L(N) = \{x1c_2c_3 \cdots c_n : x \in \{0,1\}^*, c_i \in \{0,1\}\}$$

A NFA for Text Search

- NFA accepting the set of keywords {ebay, web}



Corresponding DFA for Text Search



NFA with Epsilon Transitions - ε-NFA

- ε-NFA's allow transtions with ε label.

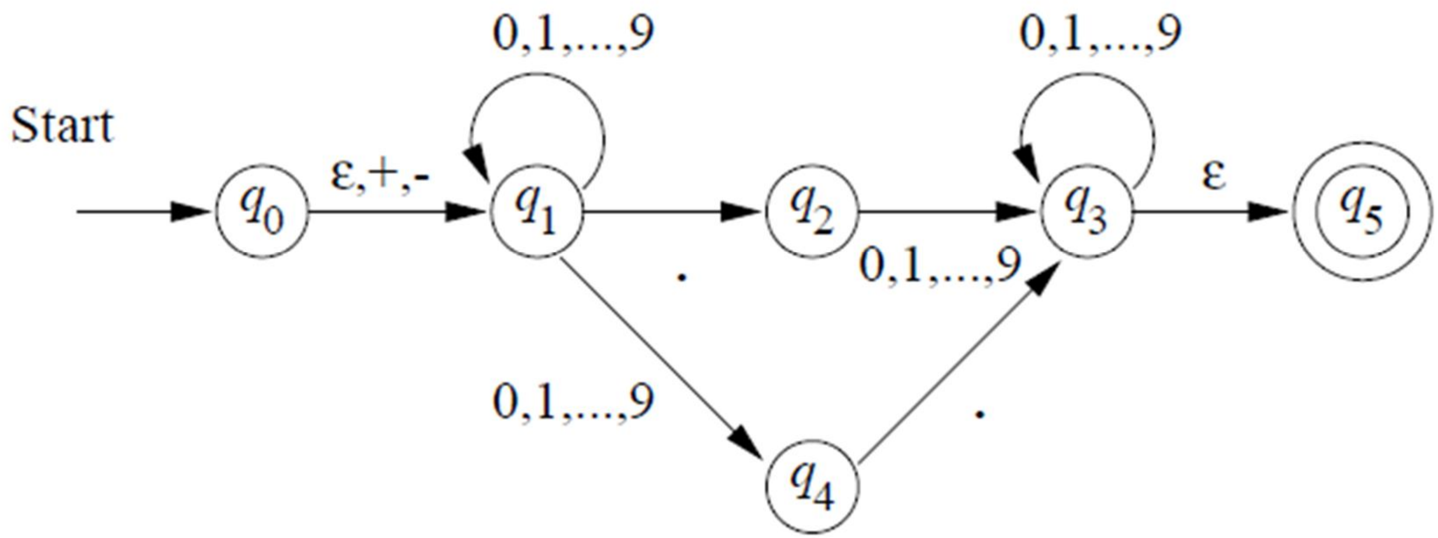
- Formally, **ε-NFA** is a quintuple

  $$\mathbf{A = (Q, \Sigma, \delta, q_0, F)}$$

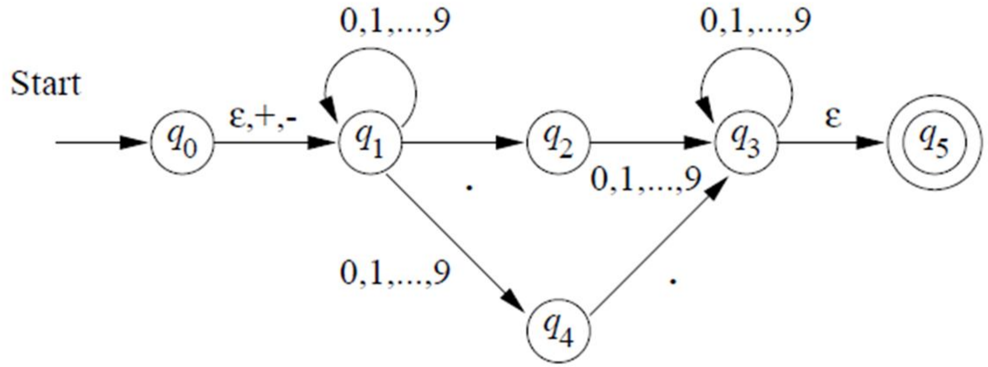
1. Q is a finite set of states
2. Σ is a finite set of symbols (alphabet)
3. Delta (δ) is a transition function from $Q \times \Sigma \cup \{\varepsilon\}$ to the powerset of Q.
4. q_0 is the start state ($q_0 \in Q$)
5. F is a set of final (accepting) states ($F \subseteq Q$)

ε-NFA Example

- An ε-NFA accepting decimal numbers consisting of:
  1. An optional + or - sign
  2. A string of digits
  3. a decimal point
  4. another string of digits
- One of the strings in (2) and .(4) are optional



ε-NFA Example - Transition Table



$$E = (\{q_0, q_1, \ldots, q_5\}, \{., +, -, 0, 1, \ldots, 9\}\ \delta, q_0, \{q_5\})$$

Transition Table

	ϵ	+,-	.	$0, \ldots, 9$
$\rightarrow q_0$	$\{q_1\}$	$\{q_1\}$	$\emptyset$	$\emptyset$
q_1	$\emptyset$	$\emptyset$	$\{q_2\}$	$\{q_1, q_4\}$
q_2	$\emptyset$	$\emptyset$	$\emptyset$	$\{q_3\}$
q_3	$\{q_5\}$	$\emptyset$	$\emptyset$	$\{q_3\}$
q_4	$\emptyset$	$\emptyset$	$\{q_3\}$	$\emptyset$
$\star q_5$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Epsilon Closure

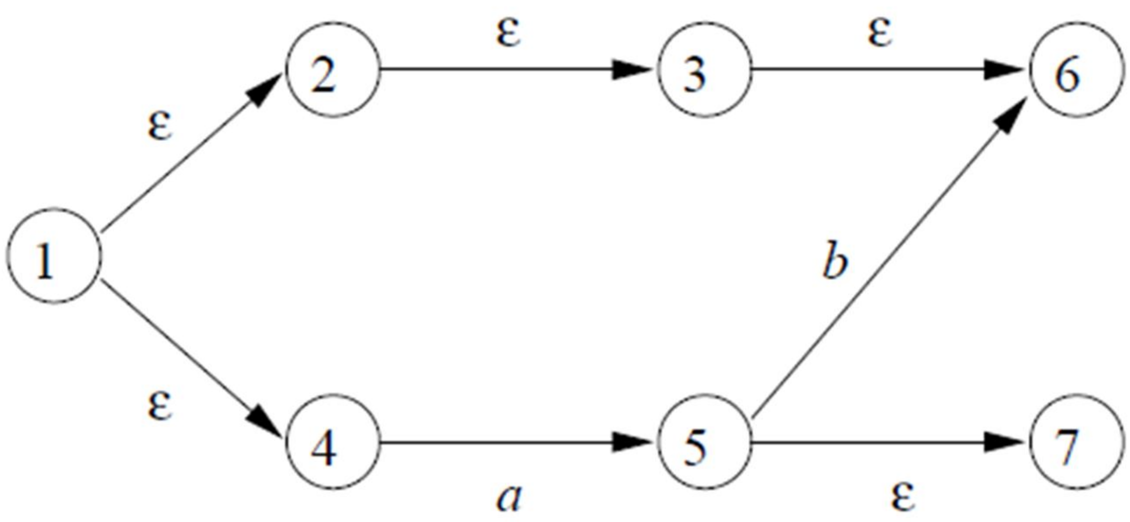
- We close a state by adding all states reachable by a sequence $\varepsilon\varepsilon \ldots \varepsilon$

- ***Inductive denition of ECLOSE(q)***

Basis: $q \in ECLOSE(q)$

Induction: $p \in ECLOSE(q)$ and $r \in \delta(p,\varepsilon) \Rightarrow r \in ECLOSE(q)$

Epsilon Closure



ECLOSE(1) = {1,2,3,4,6}

ECLOSE(2) = {2,3,6}

ECLOSE(3) = {3,6}

ECLOSE(4) = {4}

ECLOSE(5) = {5,7}

ECLOSE(6) = {6}

ECLOSE(7) = {7}

Exdended Delta for ε-NFA

- **Inductive definition $\hat{\delta}$ of for ε-NFA**

Basis: $$\hat{\delta}(q, \epsilon) = \text{ECLOSE}(q)$$

Induction: $$\hat{\delta}(q, xa) = \bigcup_{p \in \delta(\hat{\delta}(q,x),a)} \text{ECLOSE}(p)$$

Equivalence of DFA and ε-NFA

- Given an ε-NFA

$$E = (Q_E, \Sigma, \delta_E, q_0, F_E)$$

we will construct a DFA

$$D = (Q_D, \Sigma, \delta_D, q_D, F_D)$$

such that L(D) = L(E)

Equivalence of DFA and ε-NFA Subset Construction

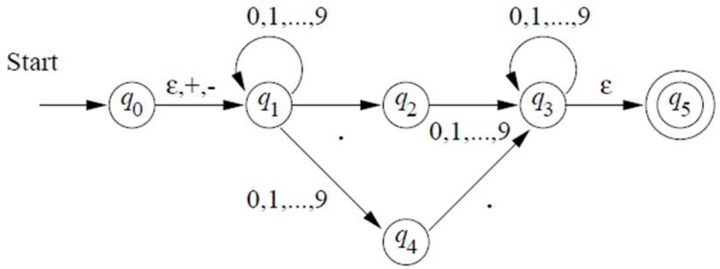
$$Q_D = \{S : S \subseteq Q_E \text{ and } S = \text{ECLOSE}(S)\}$$

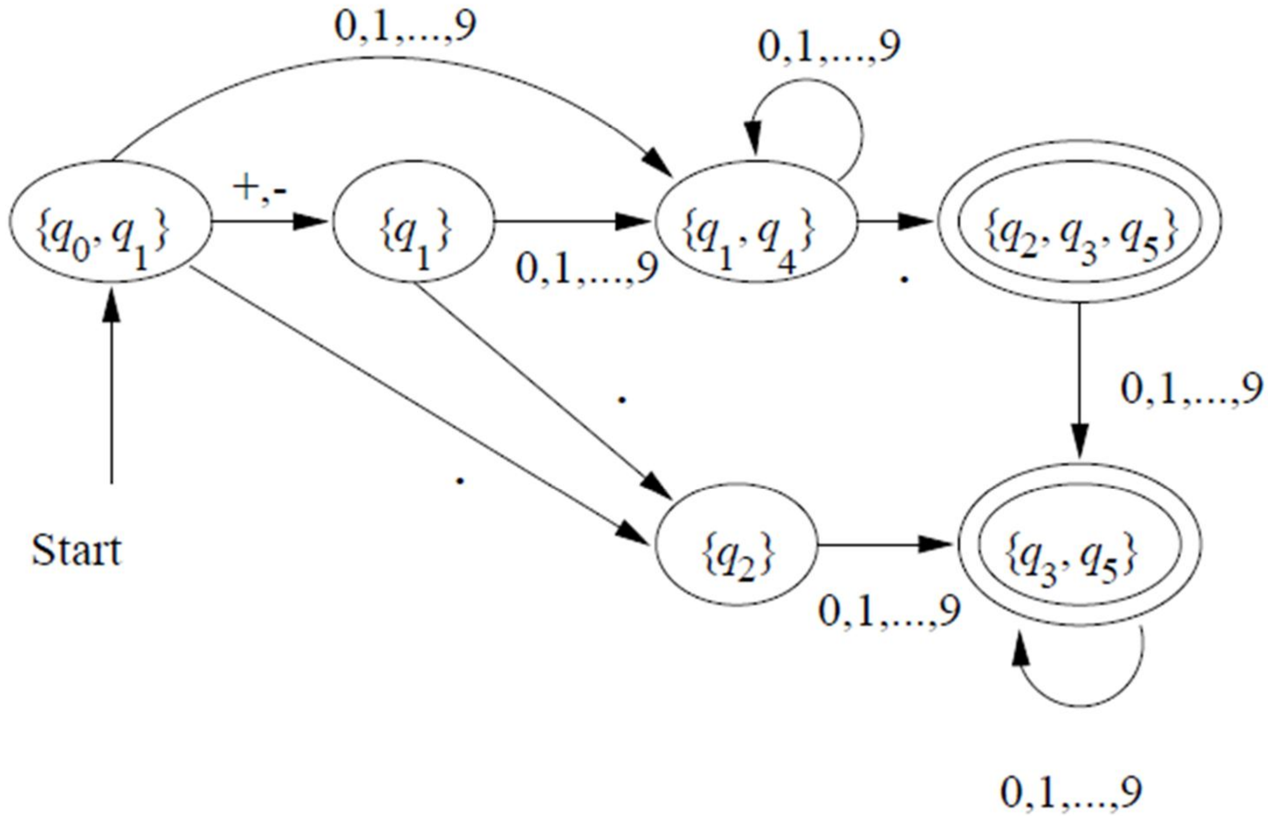
$$q_D = \text{ECLOSE}(q_0)$$

$$F_D = \{S : S \in Q_D \text{ and } S \cap F_E \neq \emptyset\}$$

$$\delta_D(S, a) = \bigcup \{\text{ECLOSE}(p) : p \in \delta(t, a) \text{ for some } t \in S\}$$

Equivalence of DFA and ε-NFA
Subset Construction - Example





Equivalence of DFA and ε-NFA - Theorem

Theorem: A language L is accepted by some ε-NFA E if and only if L is accepted by some DFA D.

Proof: We use D constructed using subset-construction and show by induction that $\hat{\delta}_D(q_0, w) = \hat{\delta}_E(q_D, w)$

Basis: $\hat{\delta}_E(q_0, \epsilon) = \mathsf{ECLOSE}(q_0) = q_D = \hat{\delta}(q_D, \epsilon)$

Induction:

$$\hat{\delta}_E(q_0, xa) = \bigcup_{p \in \delta_E(\hat{\delta}_E(q_0, x), a)} \mathsf{ECLOSE}(p)$$

$$= \bigcup_{p \in \delta_D(\hat{\delta}_D(q_D, x), a)} \mathsf{ECLOSE}(p)$$

$$= \bigcup_{p \in \hat{\delta}_D(q_D, xa)} \mathsf{ECLOSE}(p)$$

$$= \hat{\delta}_D(q_D, xa)$$