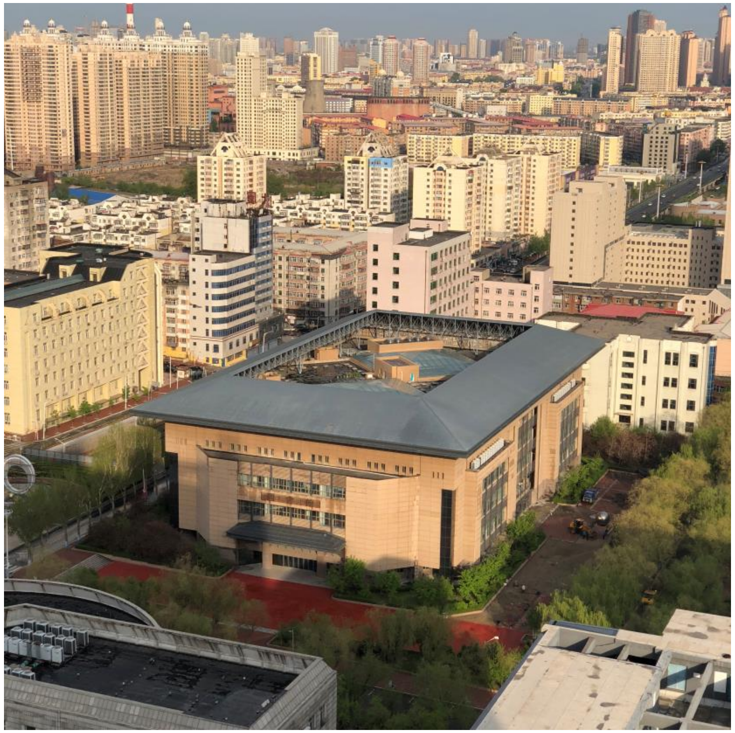
Morning

Nondeterministic Finite Automata

- Definition
- Notation
- Construction
- Language of NFA
- Equivalence with DFA

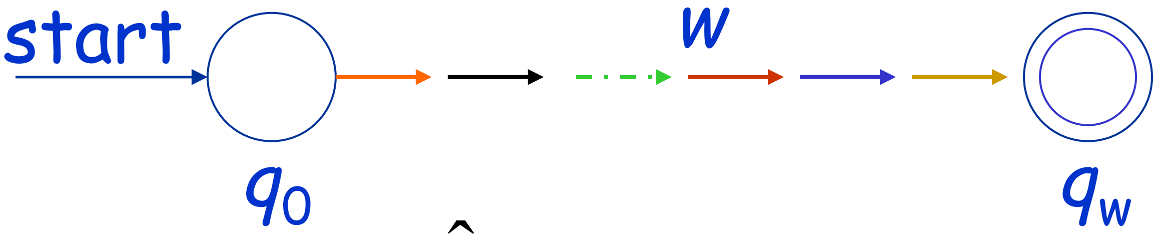


Example 1

Construct a DFA to accept

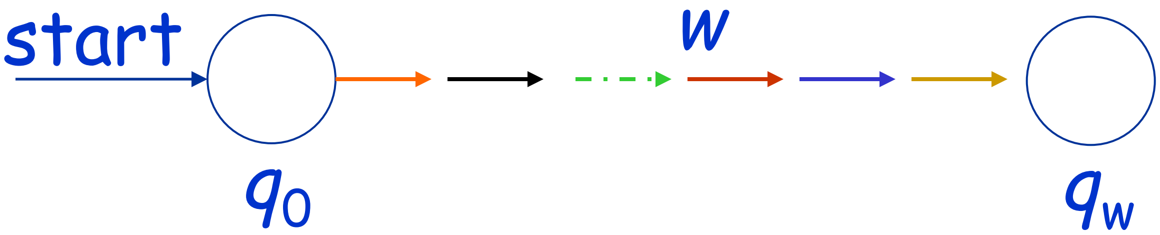
L_{x01} = {x 01 | x is any strings of 0's and 1's }

If $w \in L_{x01}$, then



$$\hat{\delta}(q, w) = q_w .$$

If $w \notin L_{x01}$, then



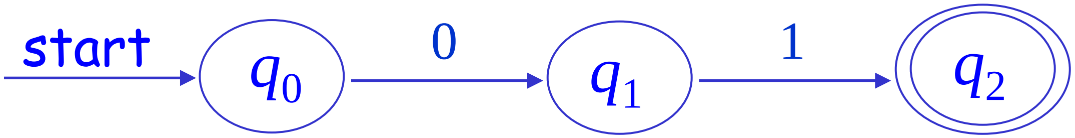
Example 1

Construct a DFA to accept

L_{x01} = {x 01 | x is any strings of 0's and 1's }

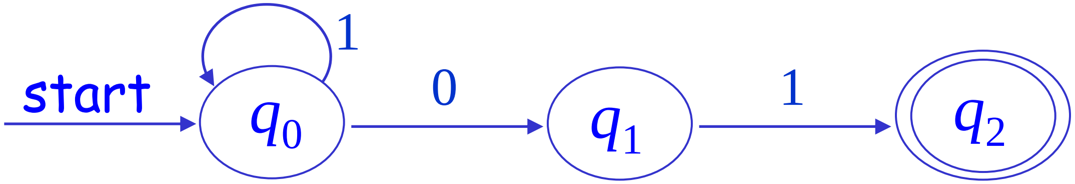
We start from the most simple string

For w=01 ,



1^*

For w=1^n01 ,
(n ≥ 0)



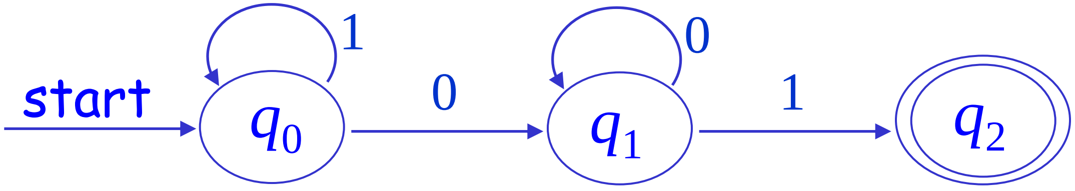
Example 1

Construct a DFA to accept

L_{x01} = {*x* 01 | *x* is any strings of 0's and 1's }

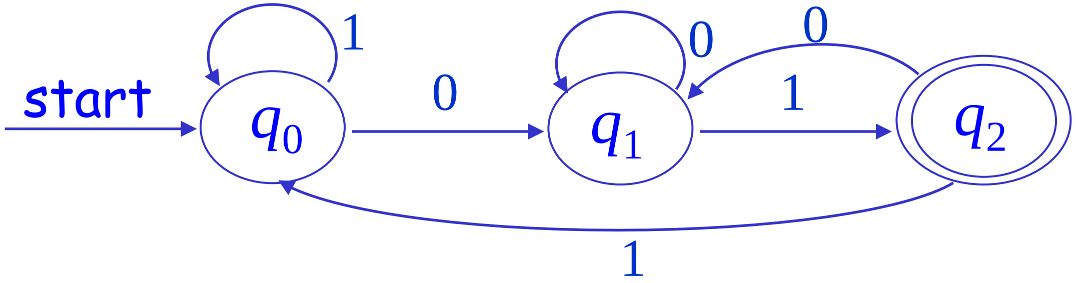
Then to more complex strings

For w=1^n00^m1 ,
(n≥ 0, m≥ 0)



Finally to the most complex strings

For w=x01 ,

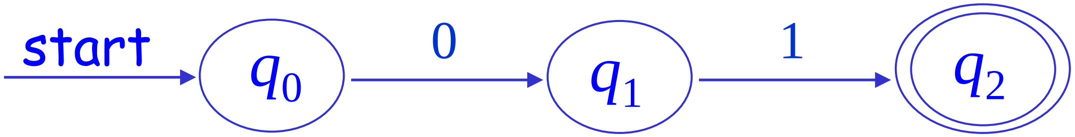


Example 1

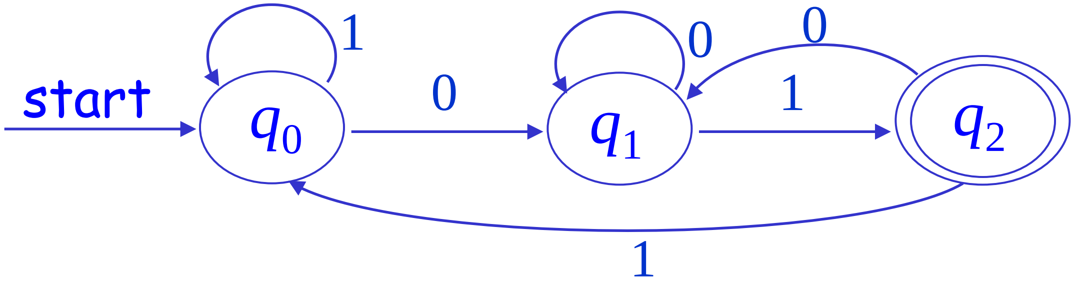
Construct a DFA to accept

L_{x01} = {x 01 | x is any strings of 0's and 1's }

Let us look at the most simple



and most complex



Formal Definition

Nondeterministic finite automaton is a five-tuple ,

such as $M = (Q, \Sigma, \delta, q_0, F)$

Where Q is a finite set of *states* ,

Σ is a finite set of *input symbols* ,

q_0 is a *start state* ,

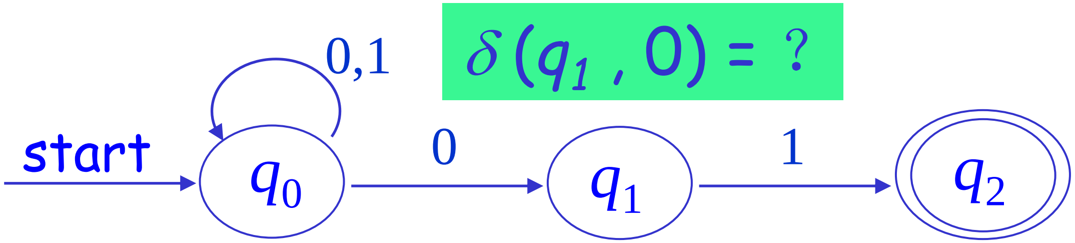
F is *a set of final state* ,

δ is *transition function* , which is a mapping

from $Q \times \Sigma$ to 2^Q.

Example 2 NFA for

L_{x01} = {x 01 | x is any strings of 0's and 1's }



$\delta(q_1, 0) = ?$

$\delta(q_2, 0) = ?$

$\delta(q_2, 1) = ?$

? x 0 1

$\delta(q_0, 0) = \{q_0, q_1\}$

Note $\delta : Q \times \Sigma \Rightarrow 2^Q$

That $\delta(q, a) = \{q_1, q_2, \dots \dots, q_n\}$

Example 2 NFA for

L_{x01} = {x 01 | x is any strings of 0's and 1's }

$N = (\{q_0 , q_1 , q_2\}, \{ 0, 1 \}, \delta , q_0 , \{q_2\})$

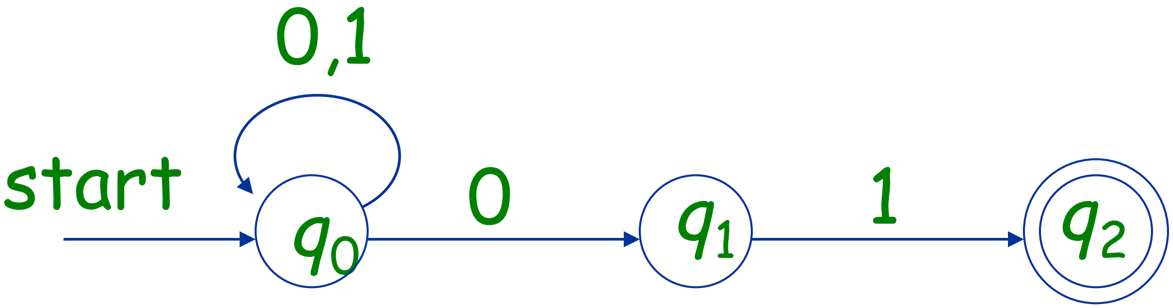
δ :

$\delta(q_0 , 0) = \{q_0 , q_1\},\quad \delta(q_0 , 1) = \{q_1\} ,$

$\delta(q_1 , 1) = \{q_2\}$

Diagram and Table Notation

Diagram

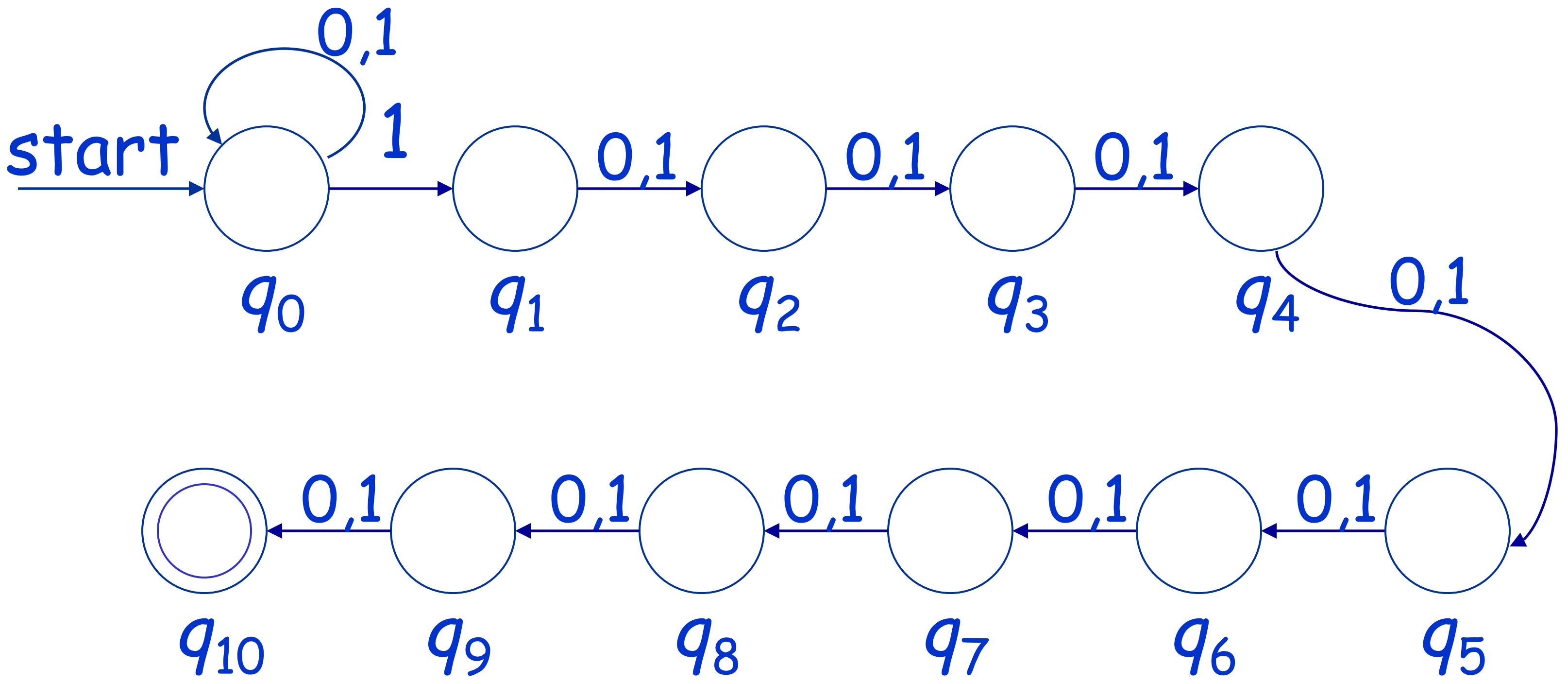


Table

	0	1
→ q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{\ \}$	$\{q_2\}$
* q_2	$\{\ \}$	$\{\ \}$

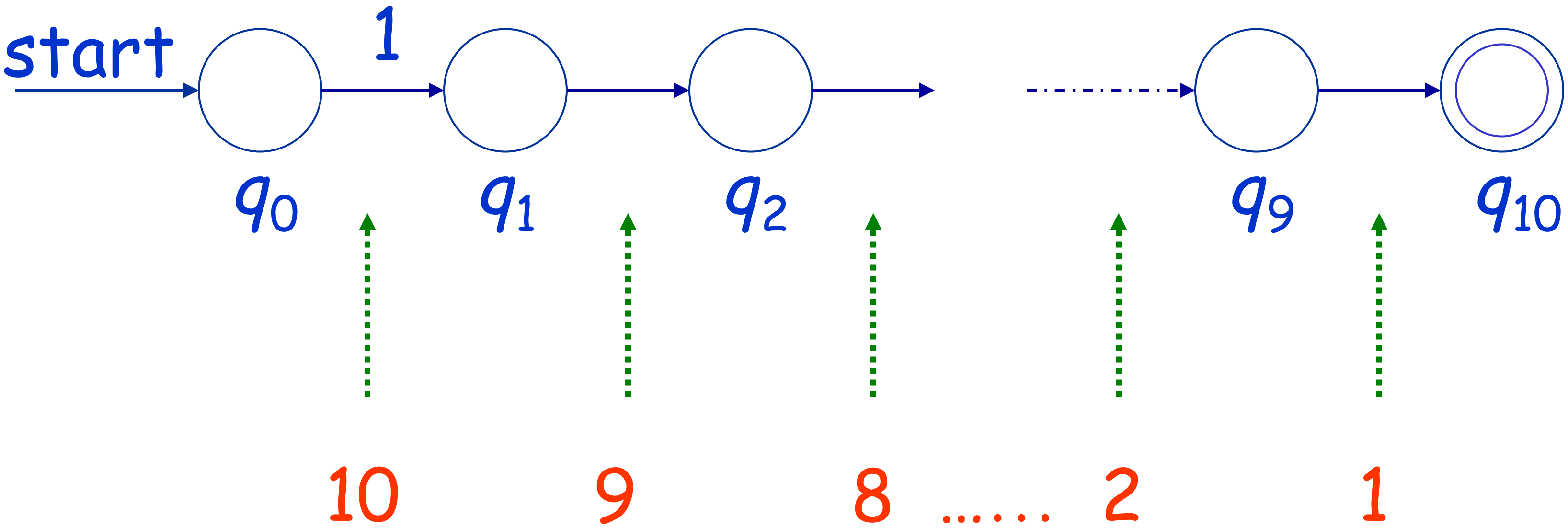
Example 3 NFA for

$L = \{w \mid w$ consists of 0's and 1's , and the tenth symbol from the right end is 1 $\}$



Shortage of DFA

L = {w | w consists of 0's and 1's , and the tenth symbol from the right end is 1 }



Extending δ to string

BASIS

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

INDUCTION

Surpose $w = xa\,,\ \hat{\delta}(q, x) = \{p_1, p_2, \cdots, p_k\}$

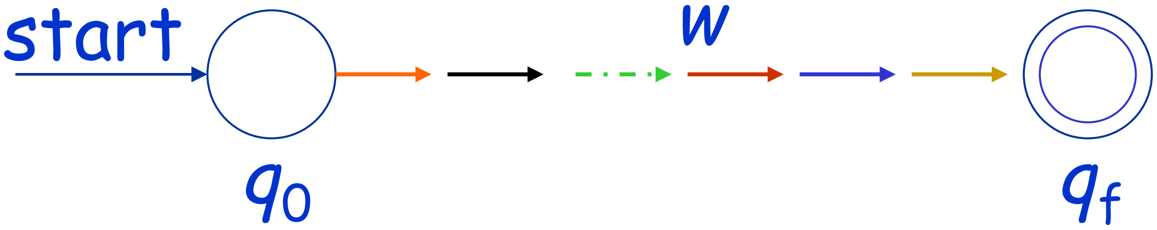
Let $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \cdots, r_m\}$

Then $\hat{\delta}(q, w) = \{r_1, r_2, \cdots, r_m\}$

Language of an NFA

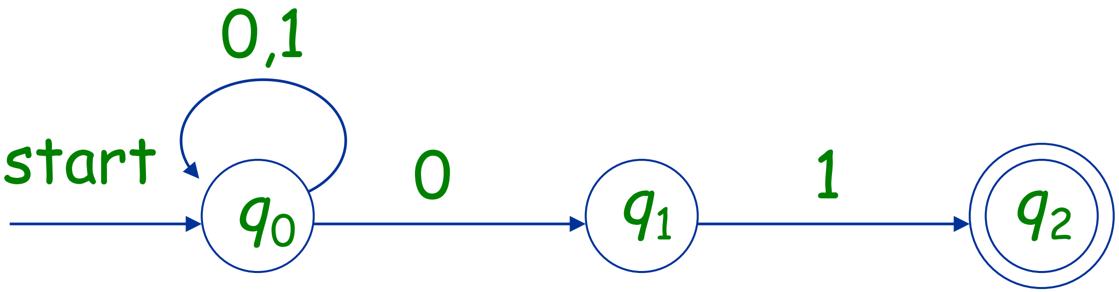
Definition The *language of an NFA* *A* is denoted *L(A)*, and defined by

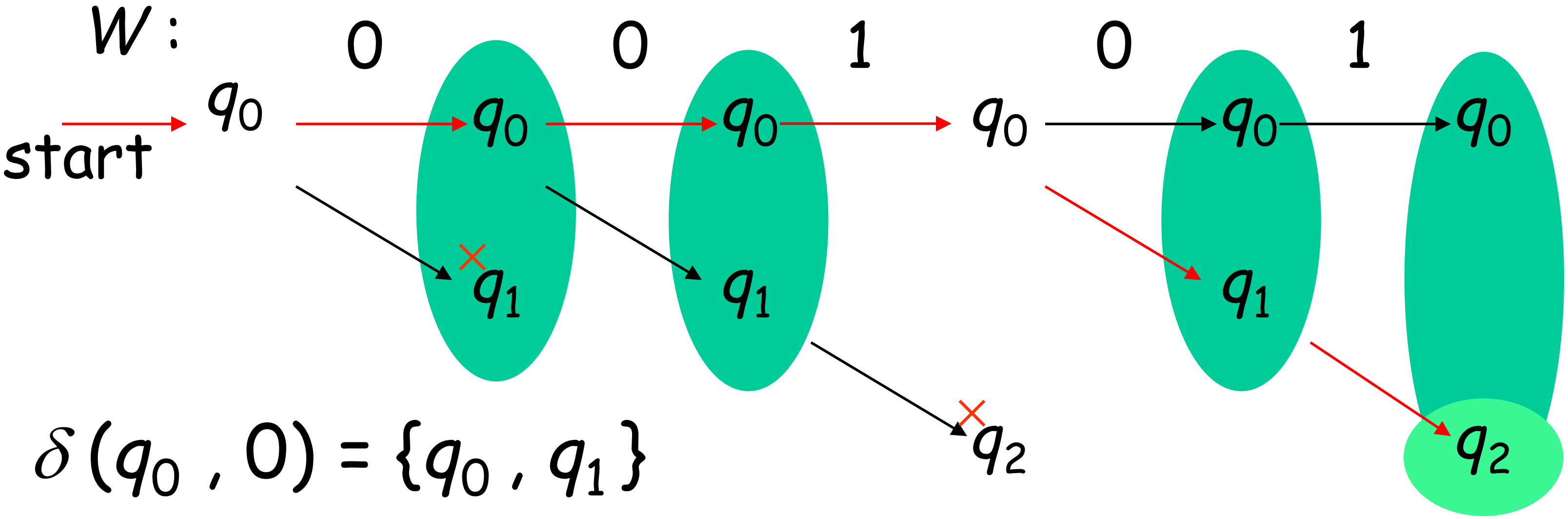
$$L(A) = \{w \mid \hat{\delta}(q_0, w) \cap F \neq \phi\}$$



There is at least a path, labeled with w, from start state to final state.

How NFA accepts a string





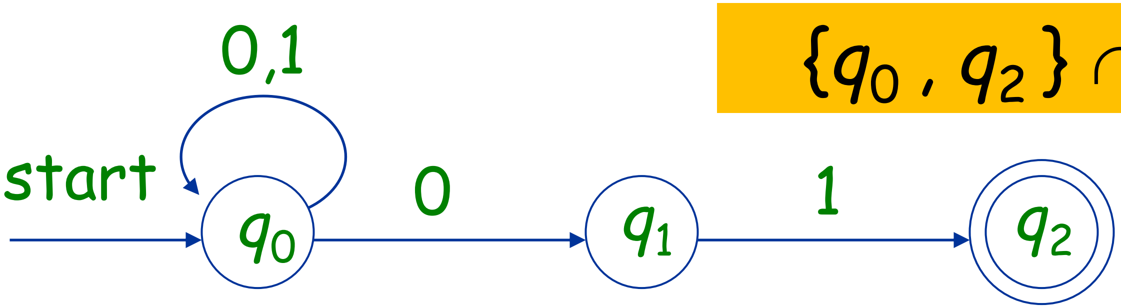
$$\delta(q_0, 0) = \{q_0, q_1\}$$

$$\delta(\{q_0, q_1\}, 0) \Rightarrow \{\delta(q_0, 0), \delta(q_1, 0)\} \Rightarrow \{q_0, q_1\}$$

$$\delta(\{q_0, q_1\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \{\ \}$$

$\hat{\delta}(q_0, w) \cap F \neq \phi$

$\{q_0, q_2\} \cap \{q_2\} \neq \phi$



Calculate $\hat{\delta}(q_0, 0\,0\,1\,0\,1)$:

$\delta(q_0, 0) = \{q_0, q_1\}$

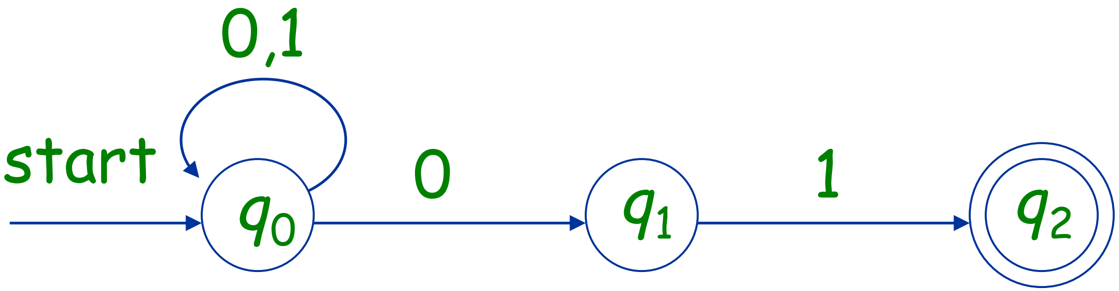
$\delta(\{q_0, q_1\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \{\ \}$

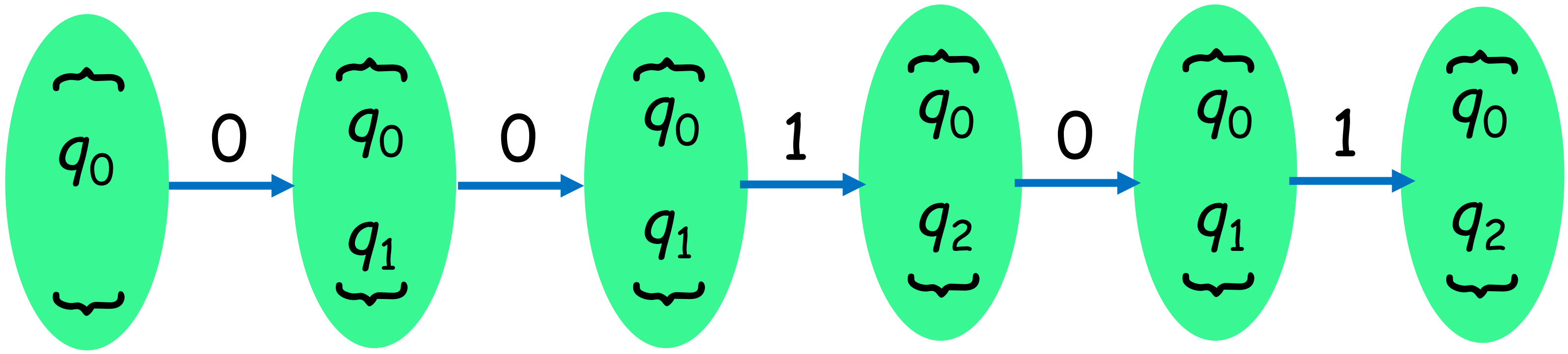
$\delta(\{q_0, q_1\}, 1) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\}$

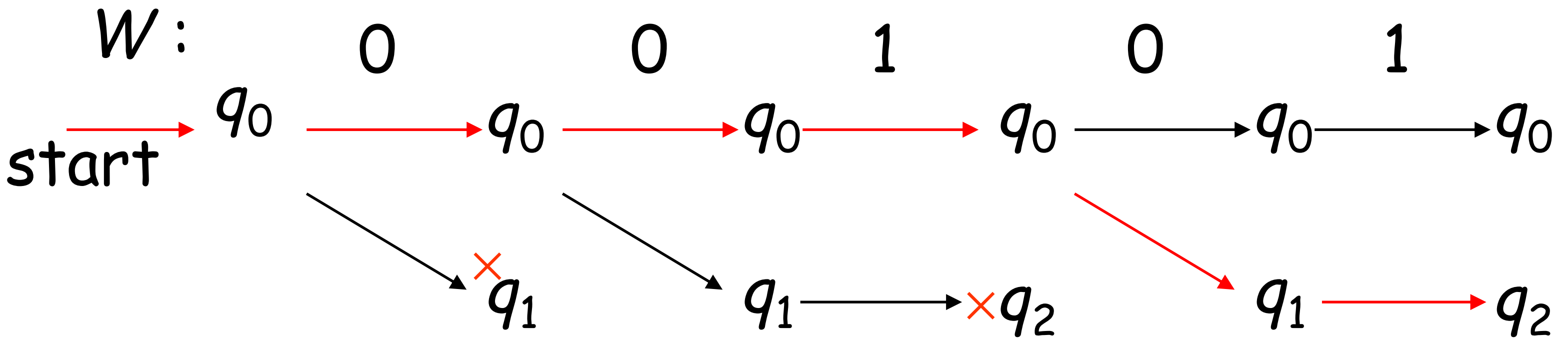
$\delta(\{q_0, q_2\}, 0) = \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\} \cup \{\ \}$

$\delta(\{q_0, q_1\}, 1) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\}$

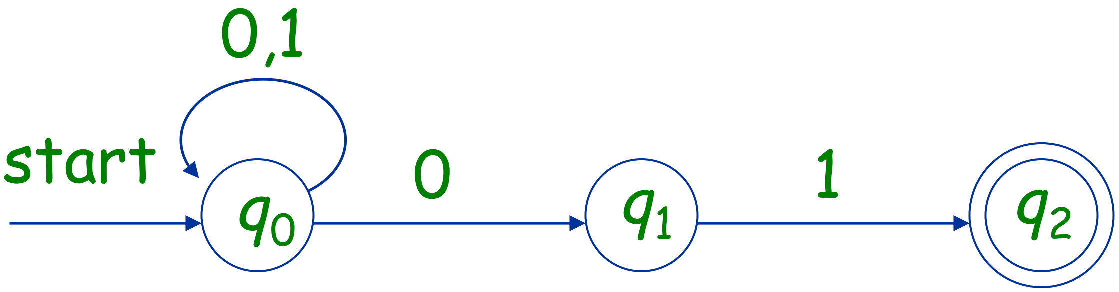
Calculate $\hat{\delta}(q_0,\ 00101)$

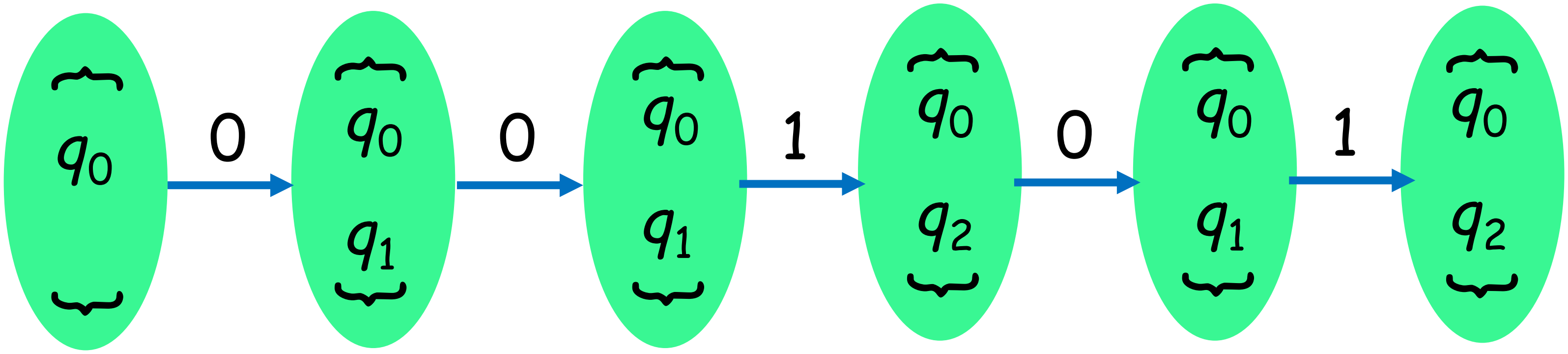


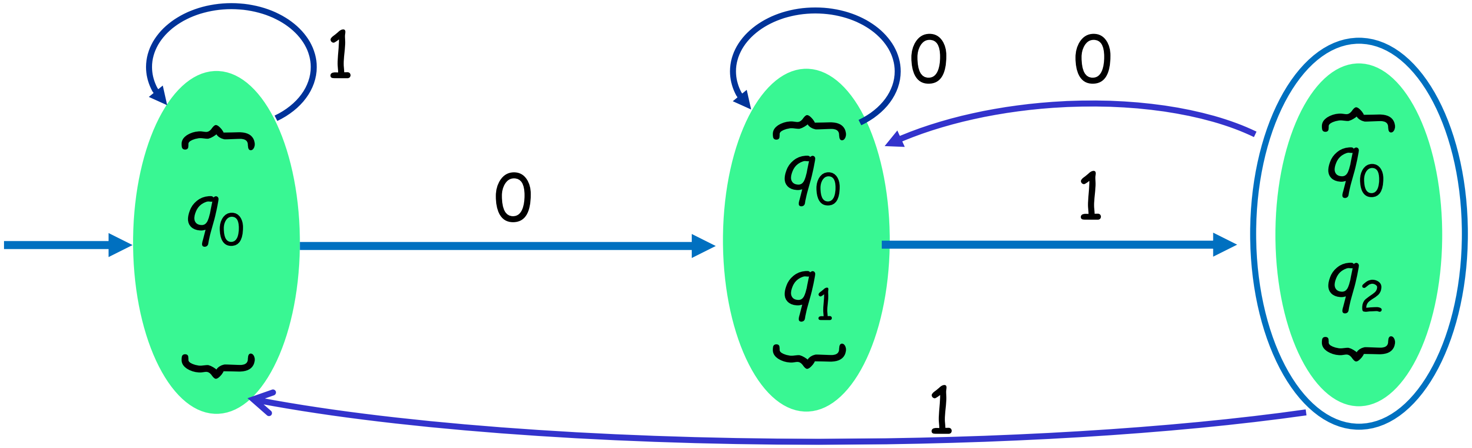




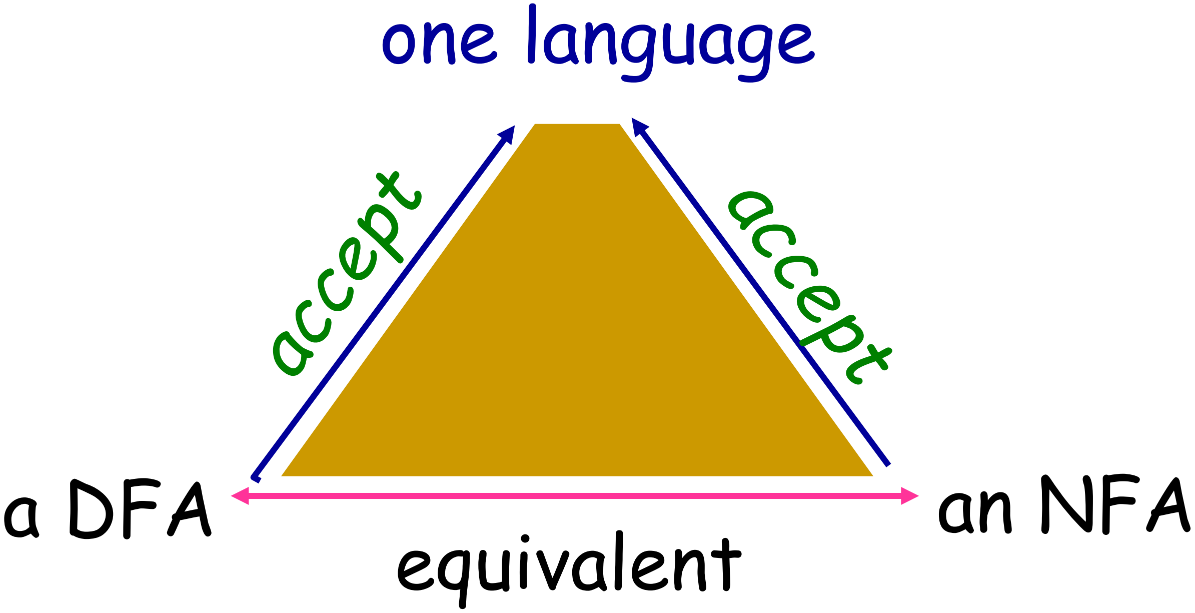
Calculate $\hat{\delta}(q_0, 00101)$







Equivalence of DFA and NFA



$$L = L(\text{NFA}) \iff L = L(\text{DFA})$$

Equivalence of DFA and NFA

Chinglish

Prove : DFA and NFA are equivalent.

- If there is an NFA accepting language L, then there must be a DFA to accept L.
- If there is a DFA accepting language L, then there must be an NFA to accept L.

- $\exists$NFA A : L=L(A) $\Rightarrow$ $\exists$DFA B : L=L(B) .
- $\exists$DFA A : L=L(A) $\Rightarrow$ $\exists$NFA B : L=L(B) .

NFA ⇒ DFA

Given an NFA : $A = (Q_N, \Sigma, \delta_N, q_0, F_N)$

Construct a DFA : $B = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$

Such that :

$$Q_D = 2^{Q_N}$$

$$2^{Q_N} = \left\{ S \mid S \in Q_N \right\}$$

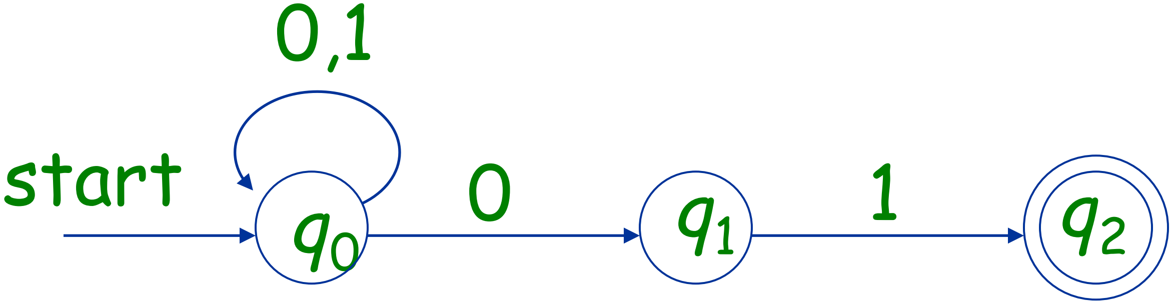
$$\delta_D(S,a) = \bigcup_{p\ in\ S} \delta_N(p,a)$$

$$F_D = \{S \mid S \subseteq Q_N \text{ } and \text{ } S \cap F_N \neq \phi\}$$

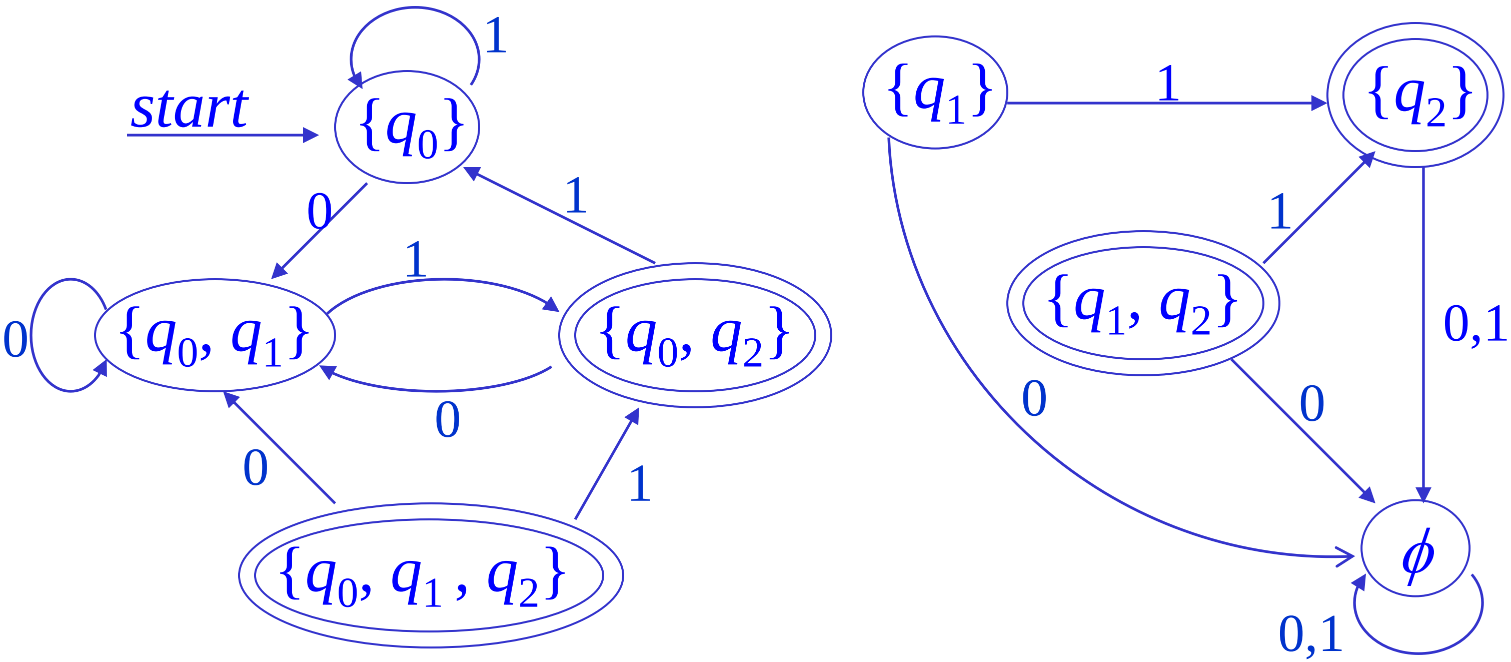
Example 4 NFA ⇒ DFA

L_{x01}={x01 | x is any strings of 0's and 1's}

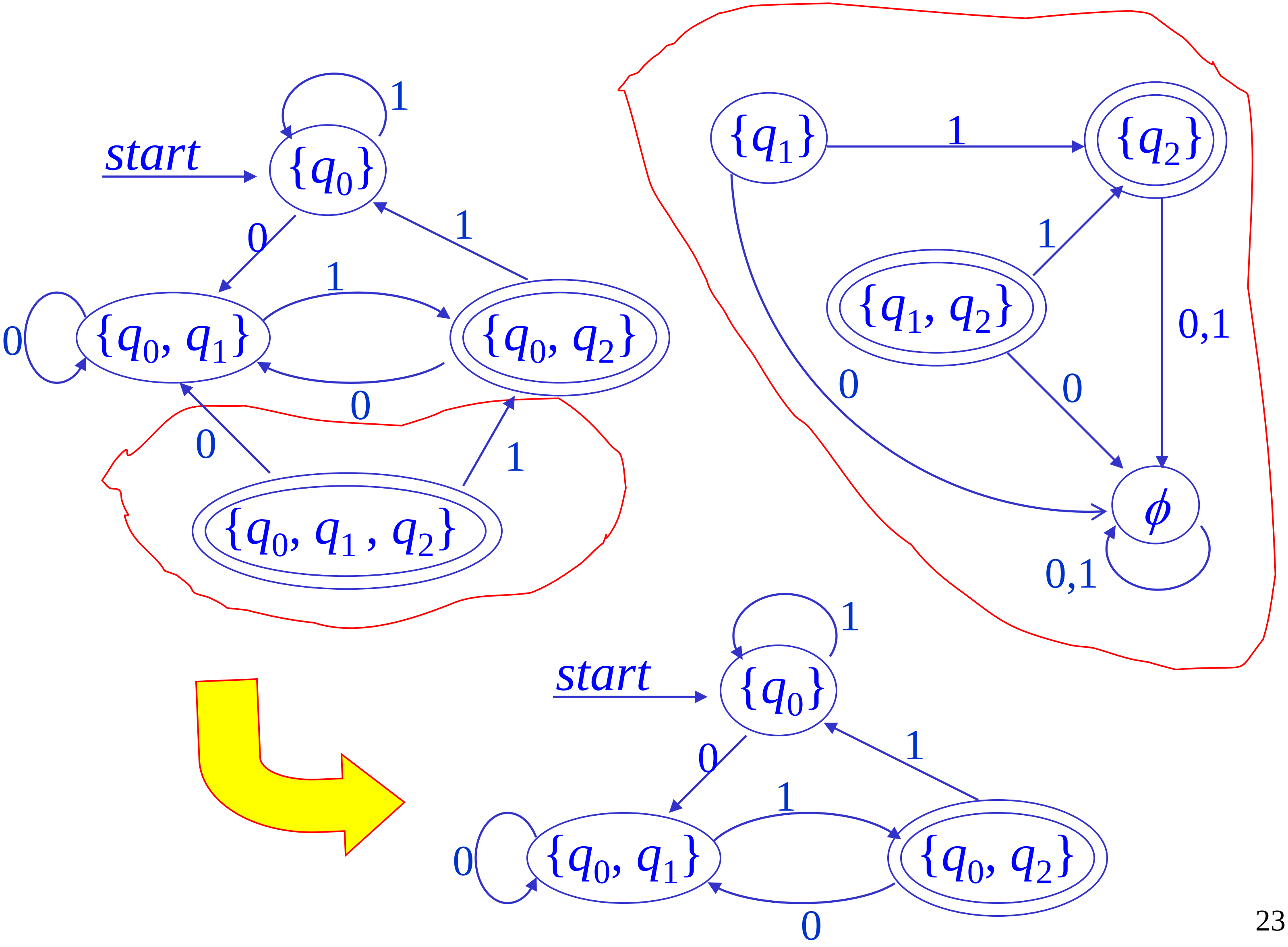
NFA :



DFA :

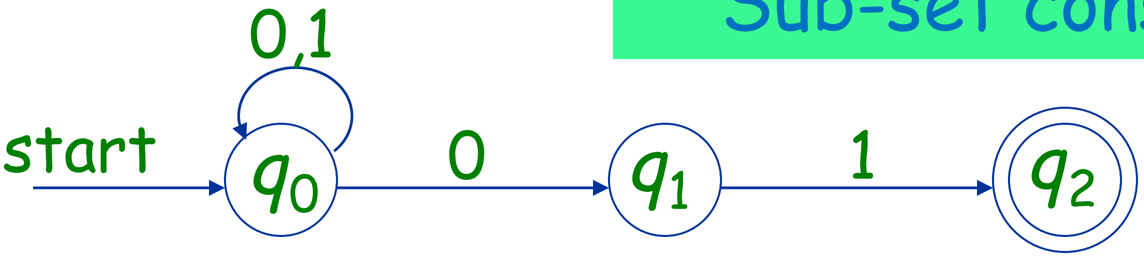


Example 4 NFA ⇒ DFA

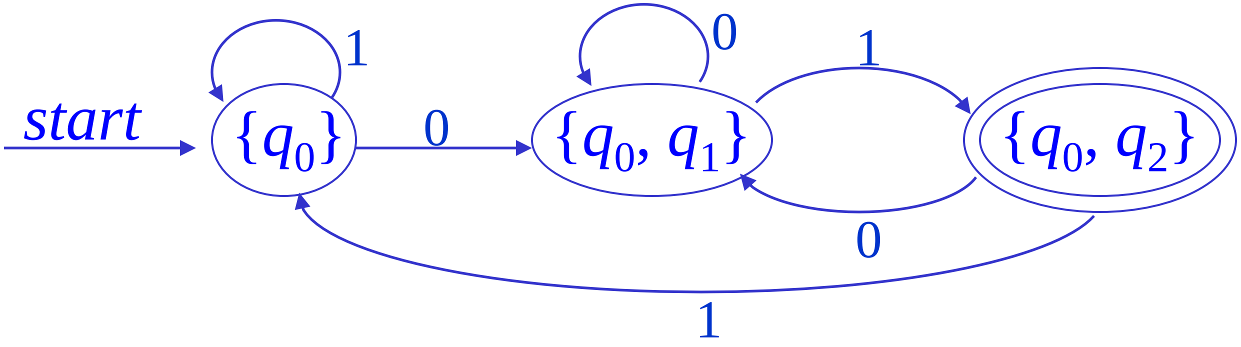


Example 4 NFA ⇒ DFA

Sub-set construction

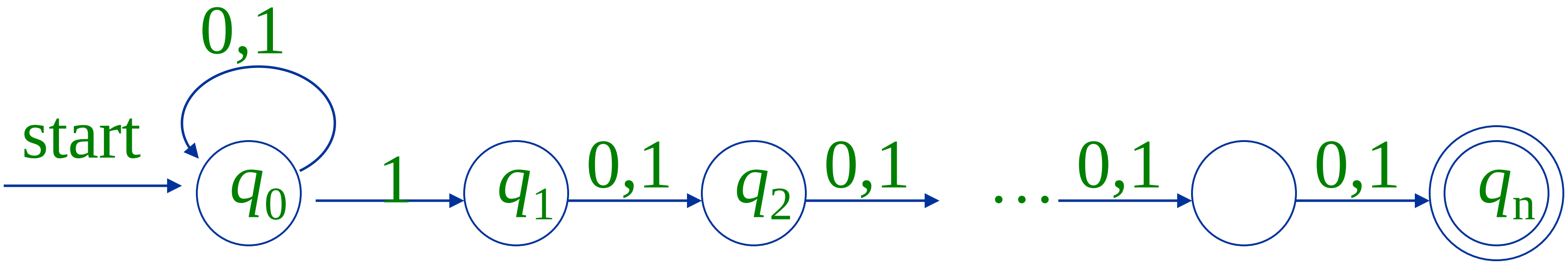


"Lazy evaluation" :



Bad case

L = {w | w consists of 0's and 1's , and the tenth symbol from the right end is 1 }



DFA ⇒ NFA

$$\text{Given a DFA}: \quad A = (\mathrm{Q}_{\mathrm{D}}, \Sigma, \delta_D, q_0, F_D)$$

$$\text{Construct an NFA}: \; N = (\mathrm{Q}_{\mathrm{N}}, \Sigma, \delta_N, q_0, F_N)$$

Such that :

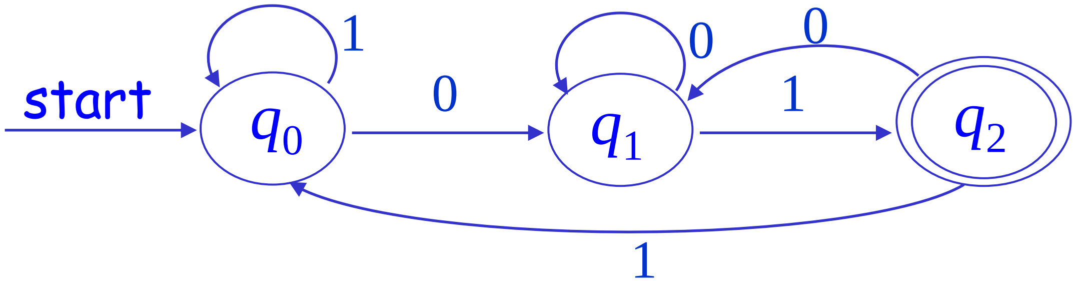
$$Q_{\mathrm{N}} = Q_D$$

$$\delta_N(q, a) = \{\delta_D(q, a)\}$$

$$F_N = F_D$$

Example 5 DFA ⇒ NFA

L_{x01} = {x 01 | x is any strings of 0's and 1's }



DFA :

$$\delta(q_0, 0) = q_1, \quad \delta(q_1, 0) = q_1, \delta(q_2, 0) = q_1$$

$$\delta(q_0, 1) = q_0, \quad \delta(q_1, 1) = q_2, \delta(q_2, 1) = q_0$$

NFA :

$$\delta(q_0, 0) = \{q_1\}, \ \delta(q_1, 0) = \{q_1\}, \delta(q_2, 0) = \{q_1\}$$

$$\delta(q_0, 1) = \{q_0\}, \delta(q_1, 1) = \{q_2\}, \delta(q_2, 1) = \{q_0\}$$

Good good study

day day up !