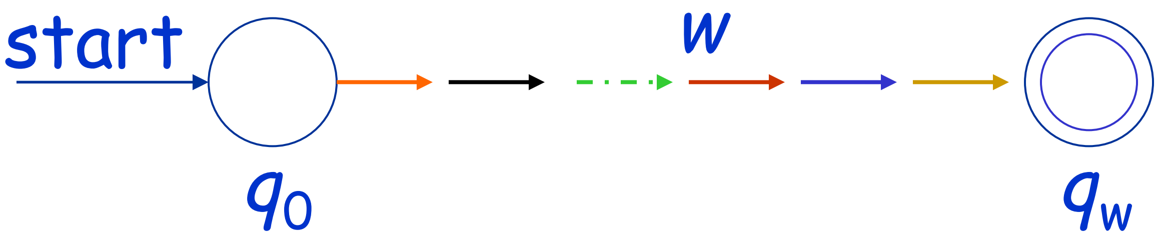
Nondeterministic Finite Automata(NFA)

1. Definition
2. Notation
3. Construction
4. Language accepted by a NFA
5. Equivalence with DFA

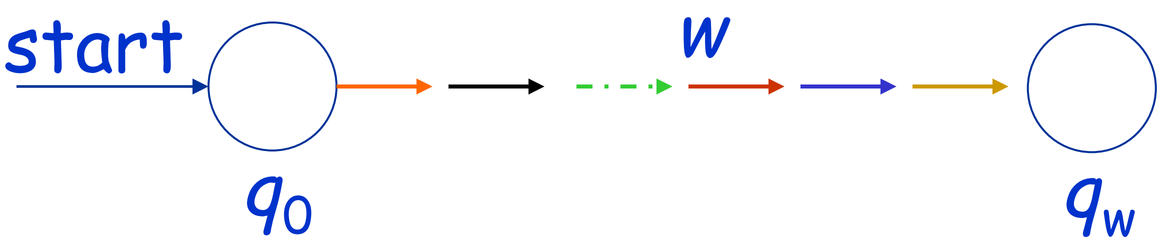
Example 3.1 Construct a DFA to accept

$$L_{x01} = \{x\,01 \mid x \text{ is any strings of 0's and 1's}\}$$

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



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

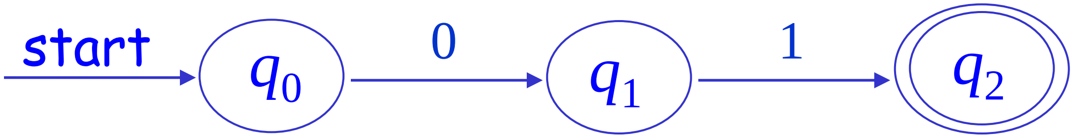


Example 3.1 Construct a DFA to accept

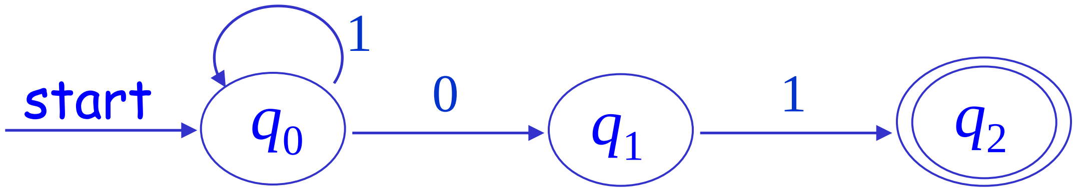
$$L_{x01} = \{x\,01 \mid x \text{ is any strings of 0's and 1's}\}$$

We start from the most simple string

For w=01 ,



For $w=1^n01$,
$(n \geq 0)$

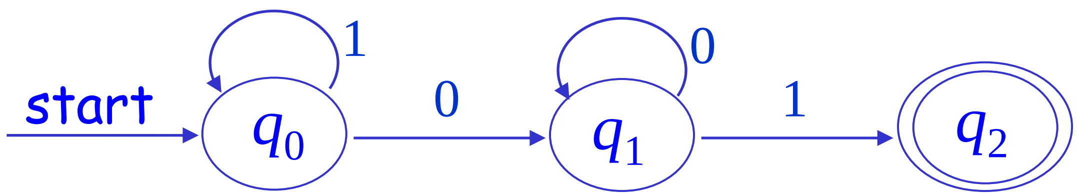


Example 3.1 Construct a DFA to accept

$$L_{x01} = \{x\,01 \mid x \text{ is any strings of 0's and 1's}\}$$

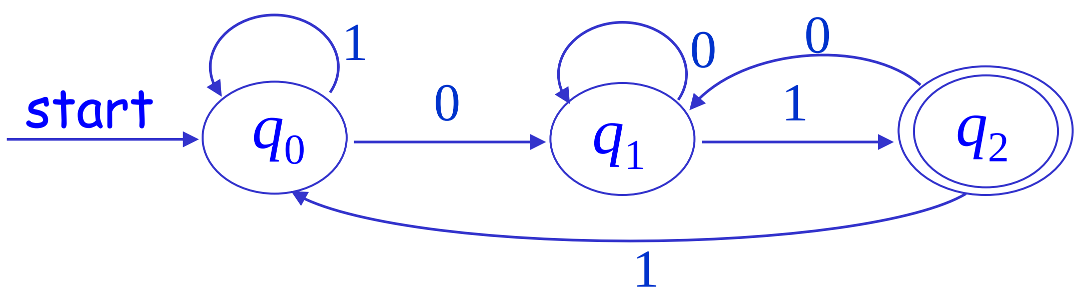
Then to more complex strings

For $w=1^n00^n1$, ($n \geq 0$)



Finally to the most complex strings

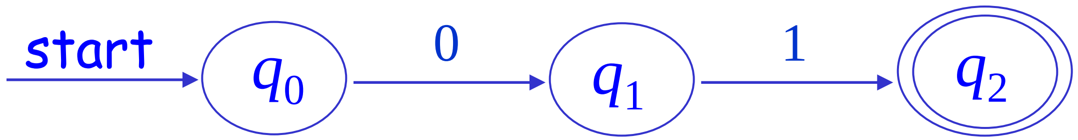
For w=x01 ,



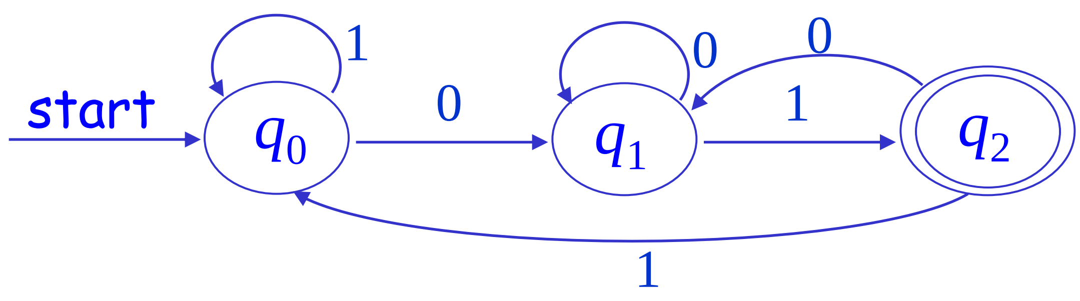
Example 3.1 Construct a DFA to accept

$$L_{x01} = \{x\,01 \mid x \text{ is any strings of 0's and 1's}\}$$

Let us look at the most simple

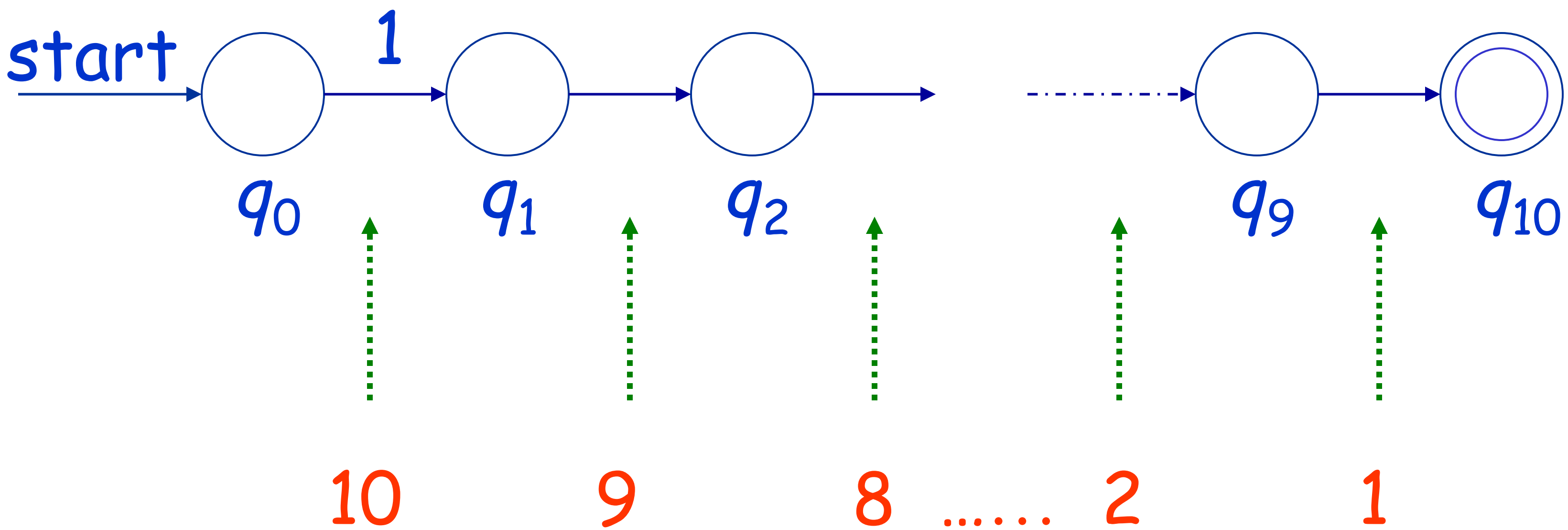


and most complex



Case for which DFA not suitable

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



Formal Definition of NFA

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 *start state* ,

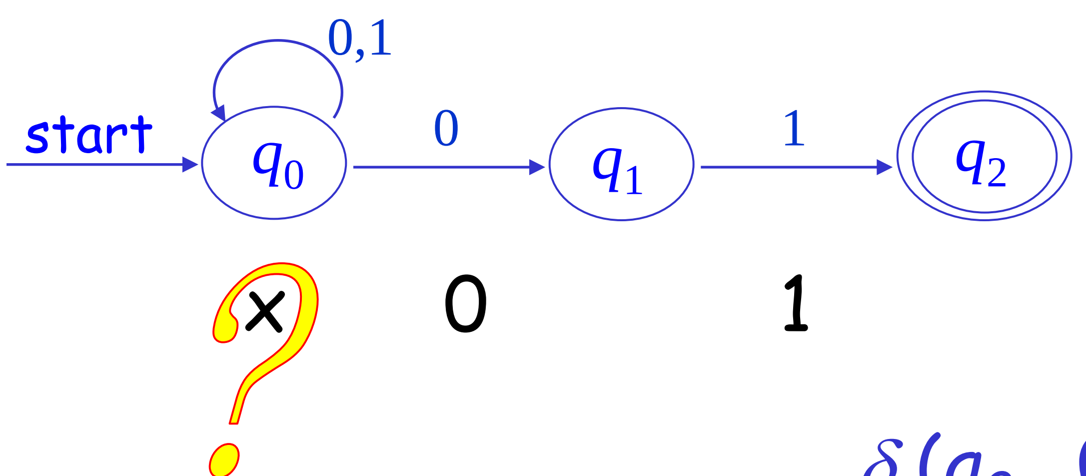
F is *a set of final state* ,

δ is *transition function* , which is a mapping

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

Example 3.2 Construct an NFA to accept

$$L_{x01} = \{x\,01 \mid x \text{ is any strings of } 0\text{'s and } 1\text{'s}\}$$



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

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

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

Example 3.2 Construct an NFA to accept

$L_{x01} = \{x\,01 \mid x$ is any strings of 0's and 1's $\}$

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

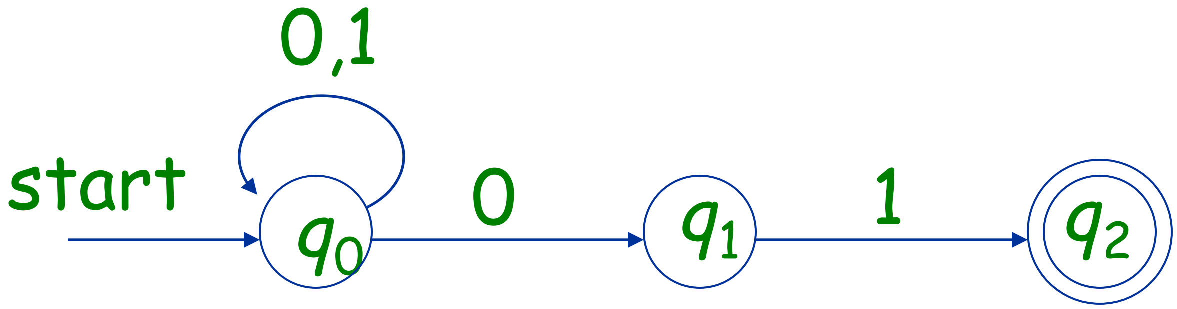
δ :

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

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

Diagram and Table Notation

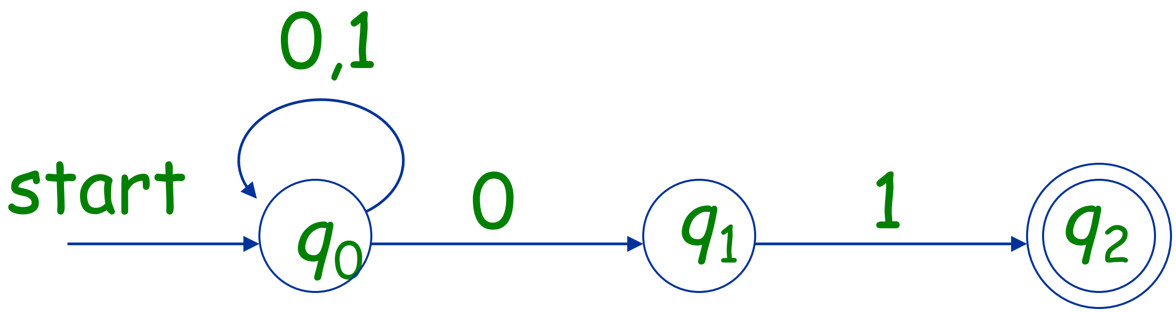
Diagram

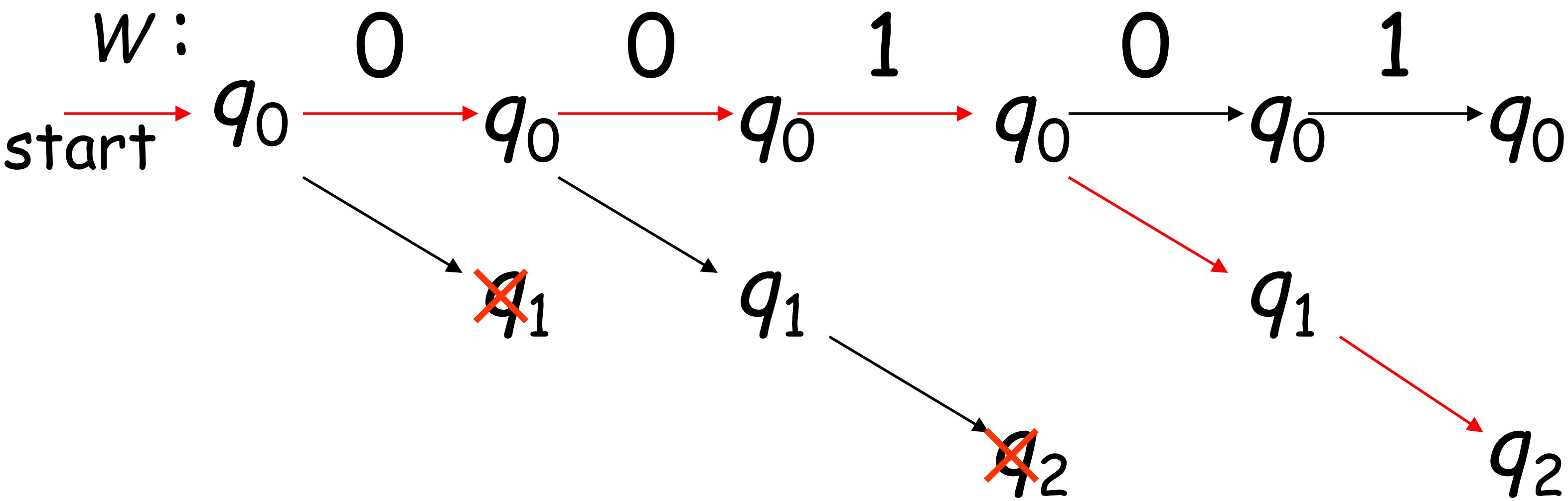


Table

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

Description

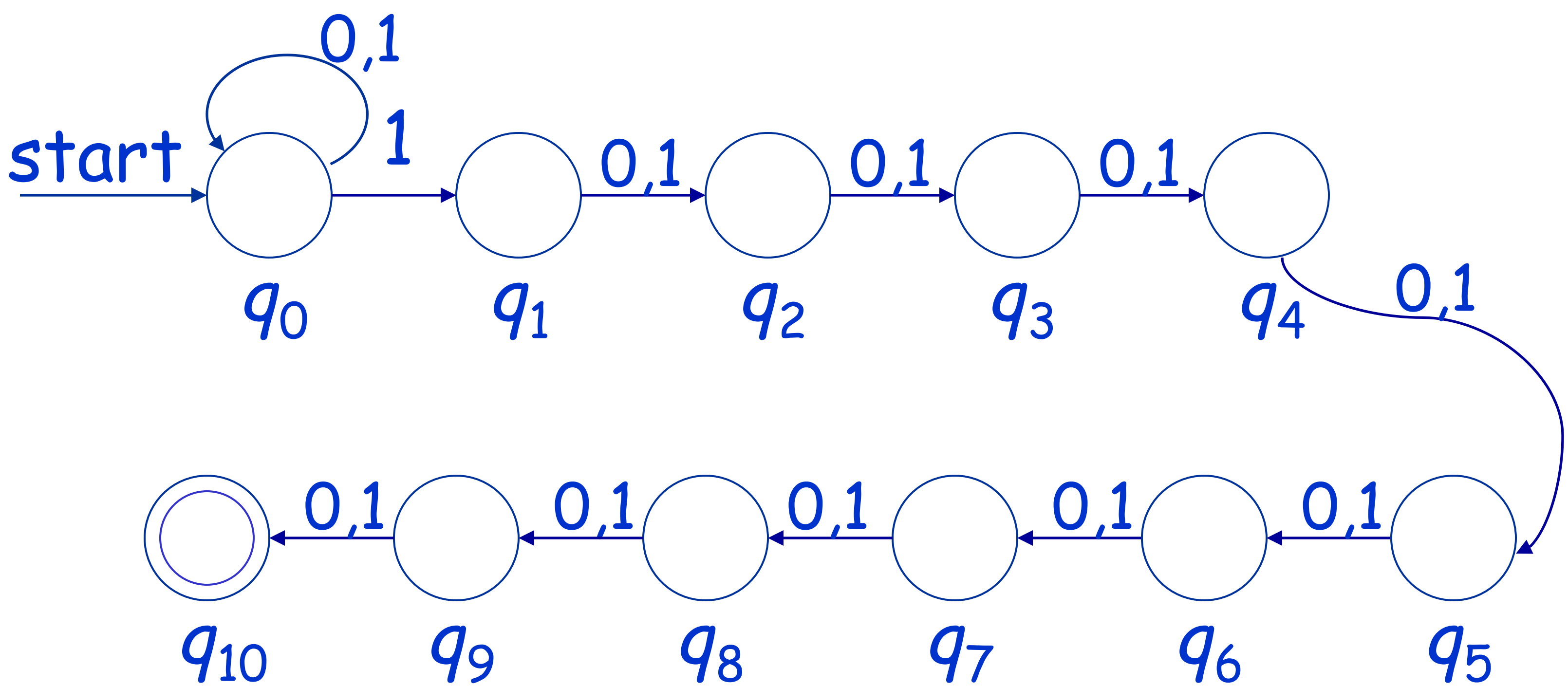




There is a path, labeled with a sequence of symbols one by one , from start state to final state.

Example 3.3 Construct an NFA to accept

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



Extending transition function to string

BASIS

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

INDUCTION

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

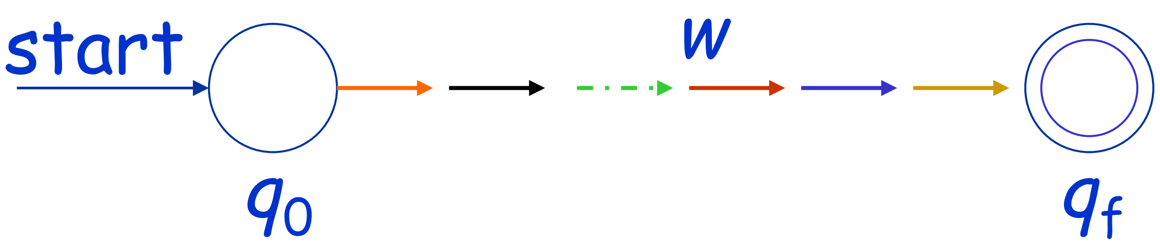
Let $\quad \bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \cdots, r_m\}$

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

The language of NFA

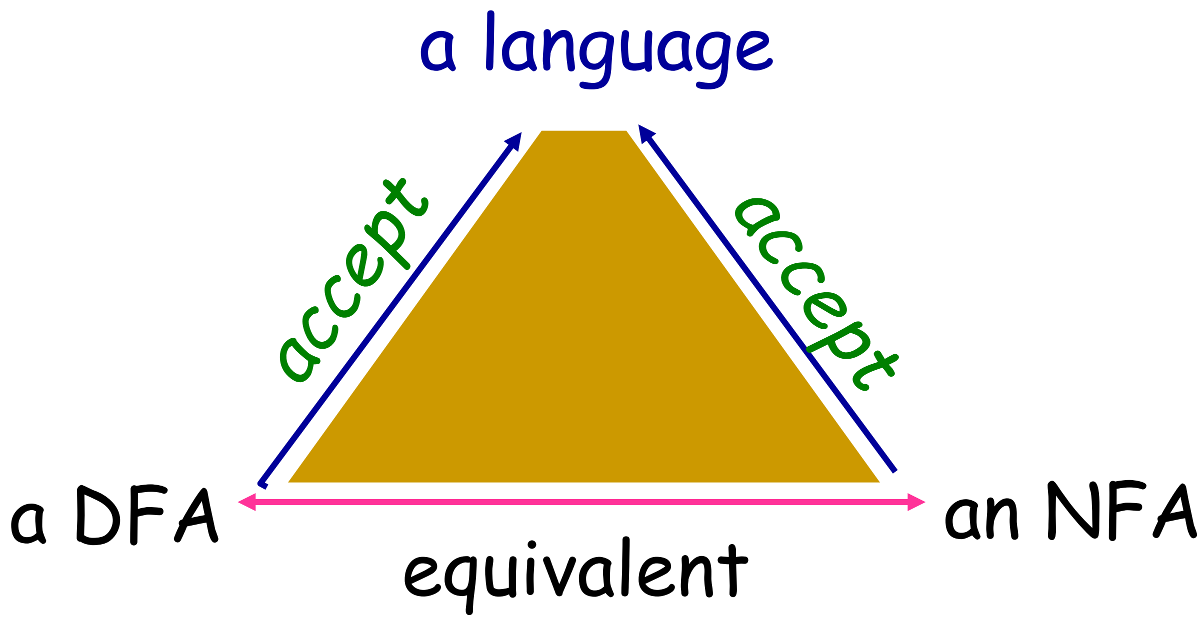
Definition The *language of an NFA* A is denoted $L(A)$, and defined by

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



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

Equivalence of DFA and NFA



If a DFA and an NFA accepts the same language , then we say that they are equivalent.

Equivalence : NFA $\Rightarrow$ DFA

Given an NFA: $N = (\mathrm{Q_N}, \Sigma, \delta_N, q_0, F_N)$

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

Such that :

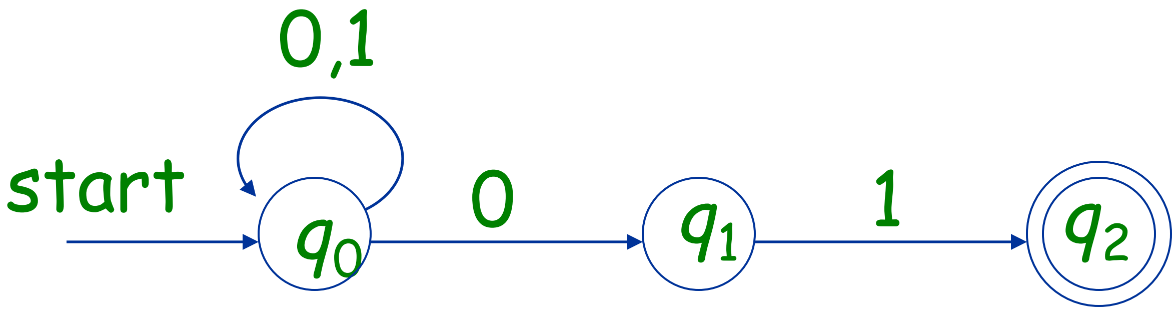
$$\mathrm{Q_D} = 2^{Q_N}$$

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

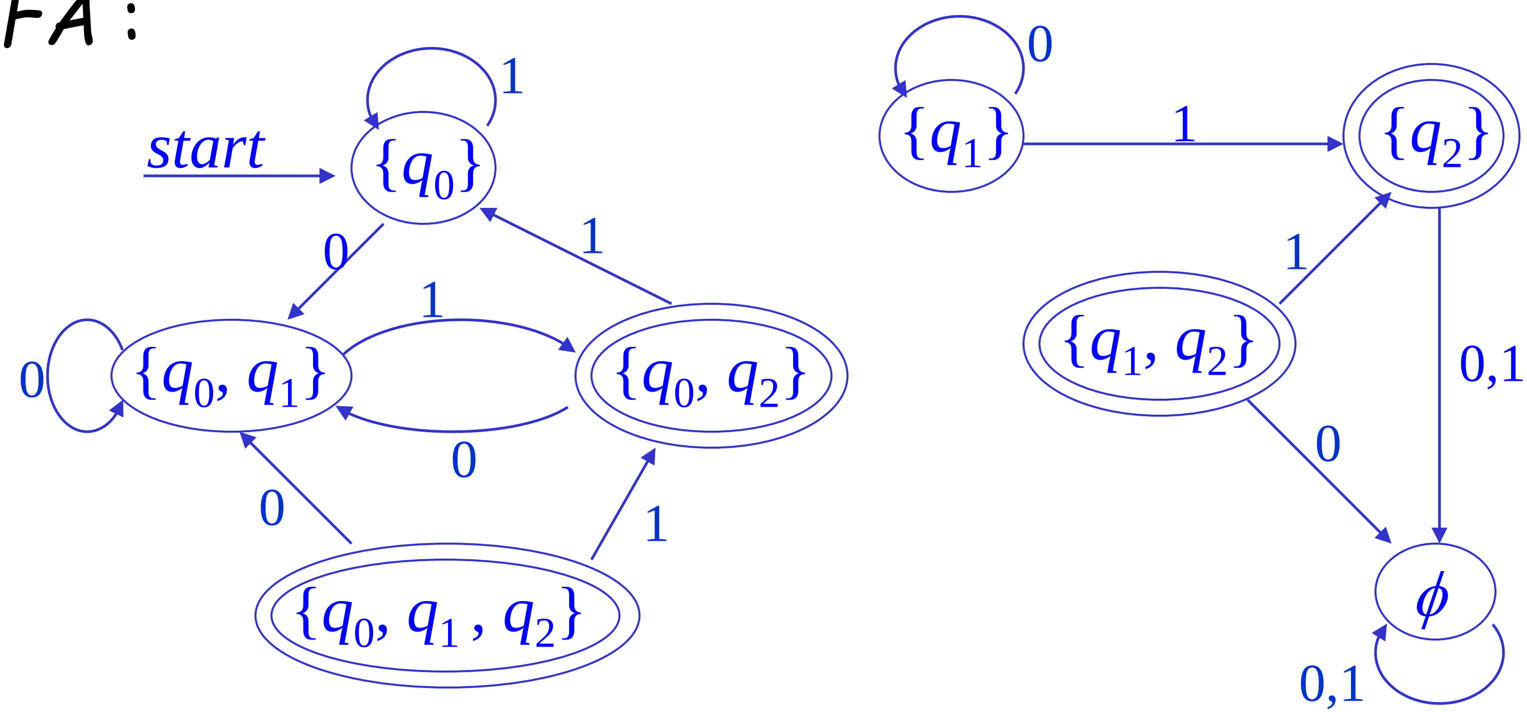
$$F_D = \{S \mid S \subseteq Q_N \ and \ S \cap F_N \neq \phi\}$$

Example 4.4

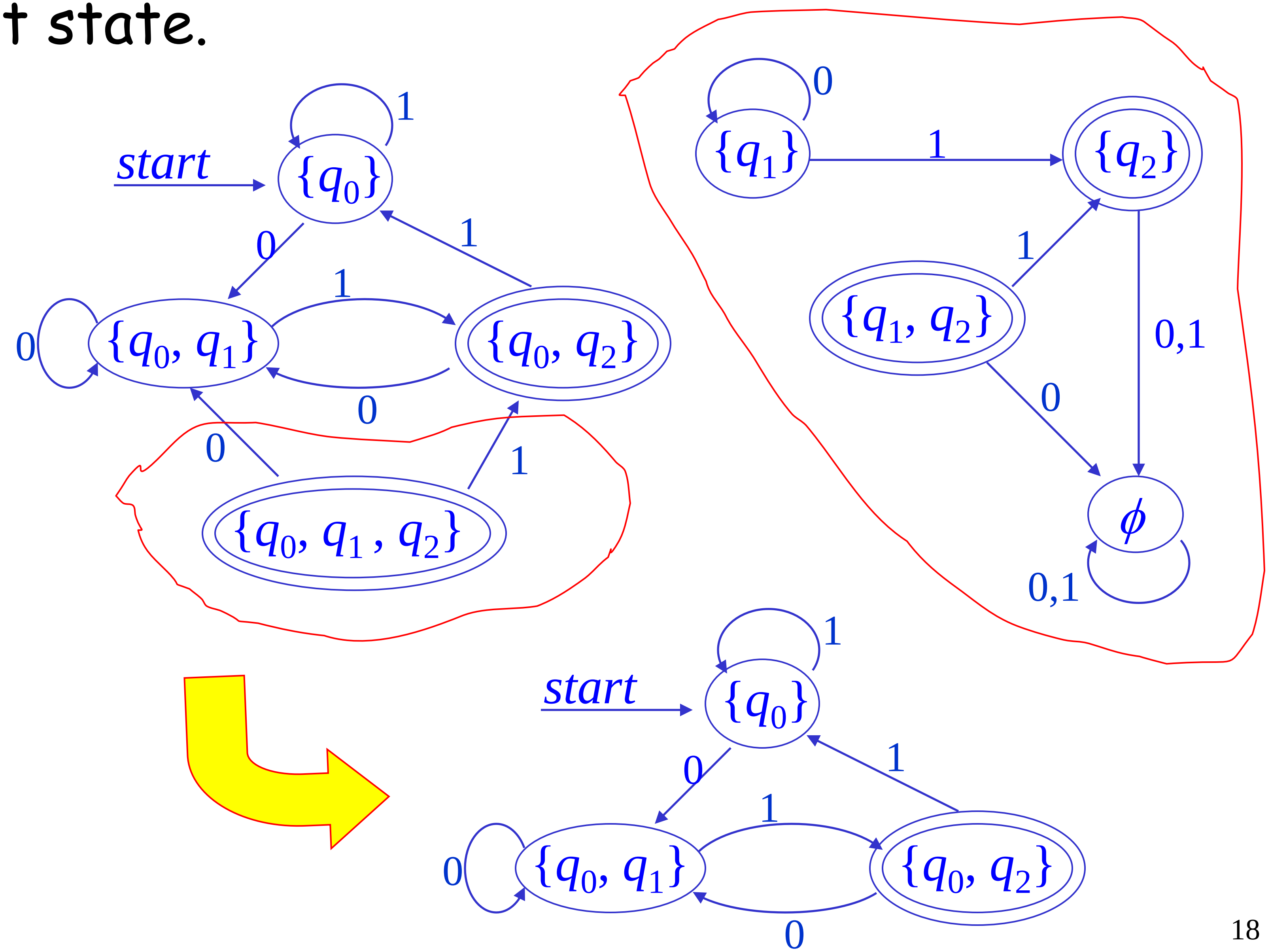
L_{x01}={x01 | x is any strings of 0's and 1's}

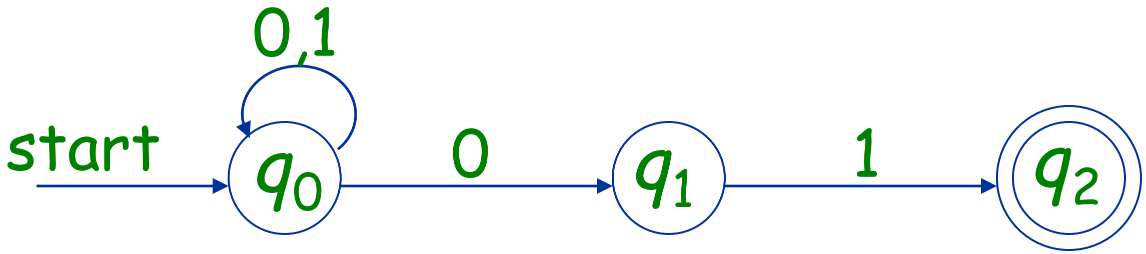


DFA :

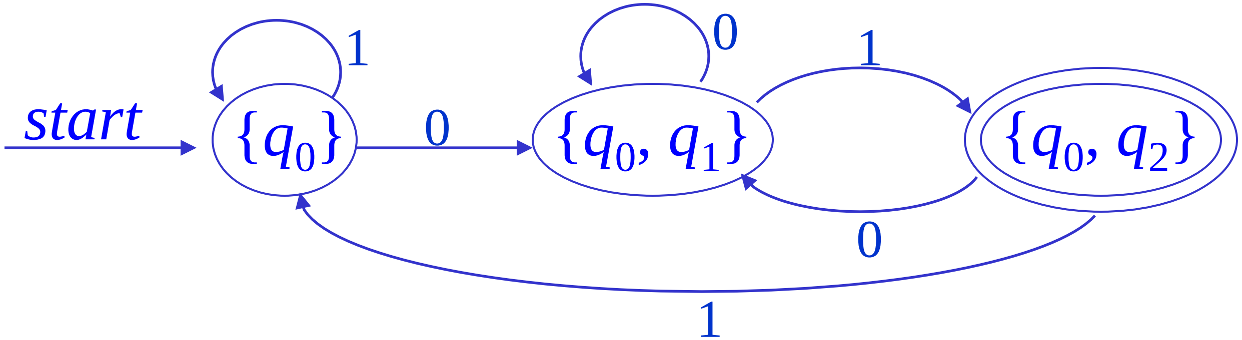


Eliminate the states which can't be reached from start state.



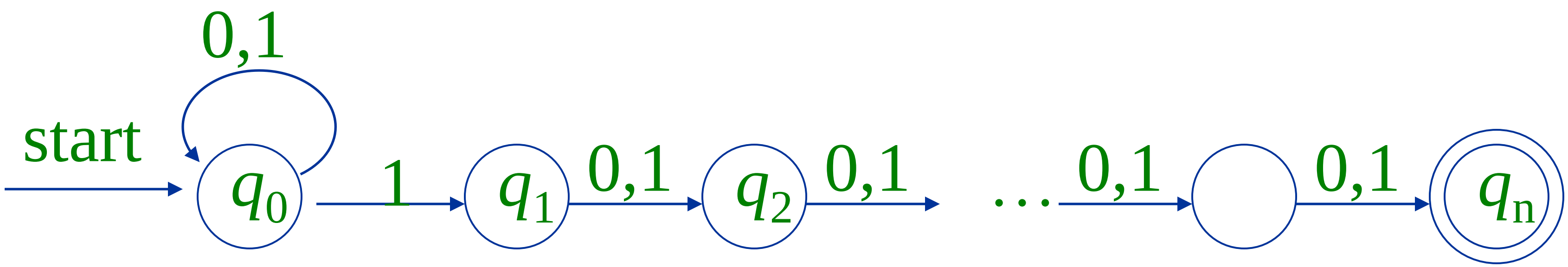


"Lazy evaluation" :



Bad case

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



Equivalence : DFA $\Rightarrow$ NFA

Given a DFA: $A = (Q_D, \Sigma, \delta_D, q_0, F_D)$

Construct an NFA: $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$

Such that :

$$Q_N = Q_D$$

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

$$F_N = F_D$$

Good good study

day day up !