Morning

Formal Definition of ε-NFA

An NFA with ε transition 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 s* ,

q_0 is *start state* ,

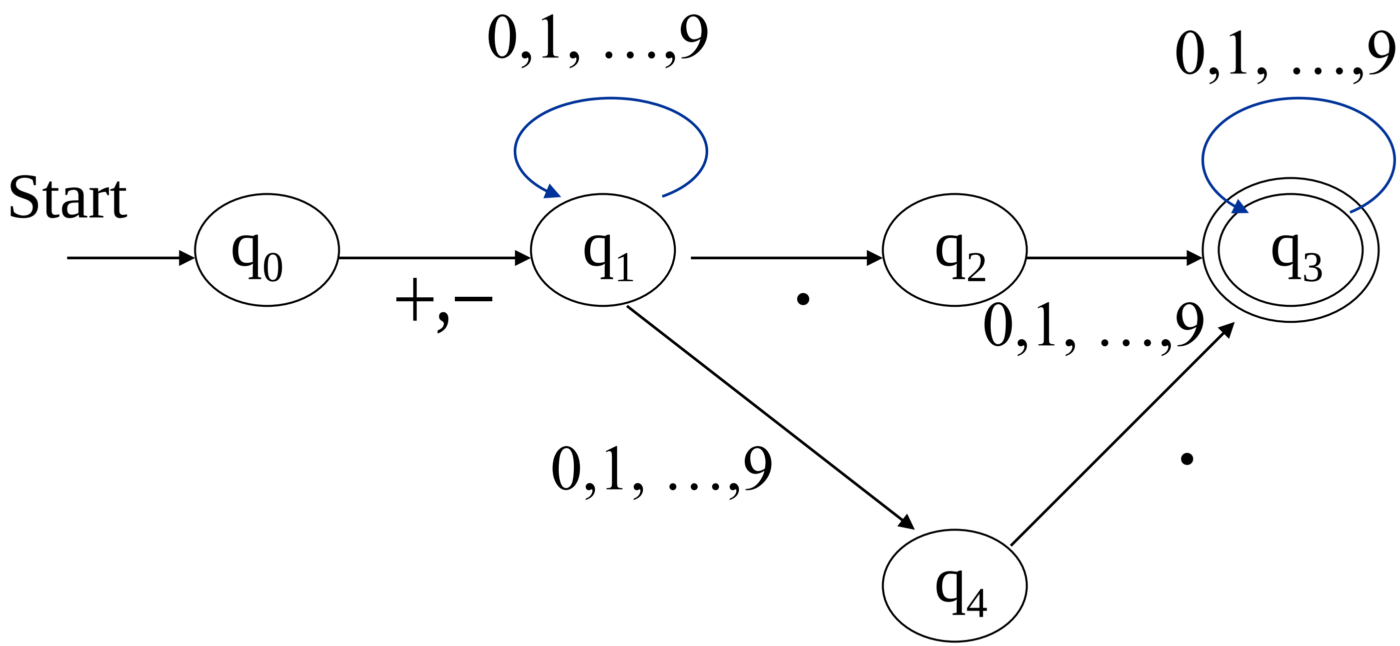
F is *a set of final state* ,

δ is *transition function* , which is a mapping

from $Q \times (\Sigma \cup \{\varepsilon\})$ to 2^Q .

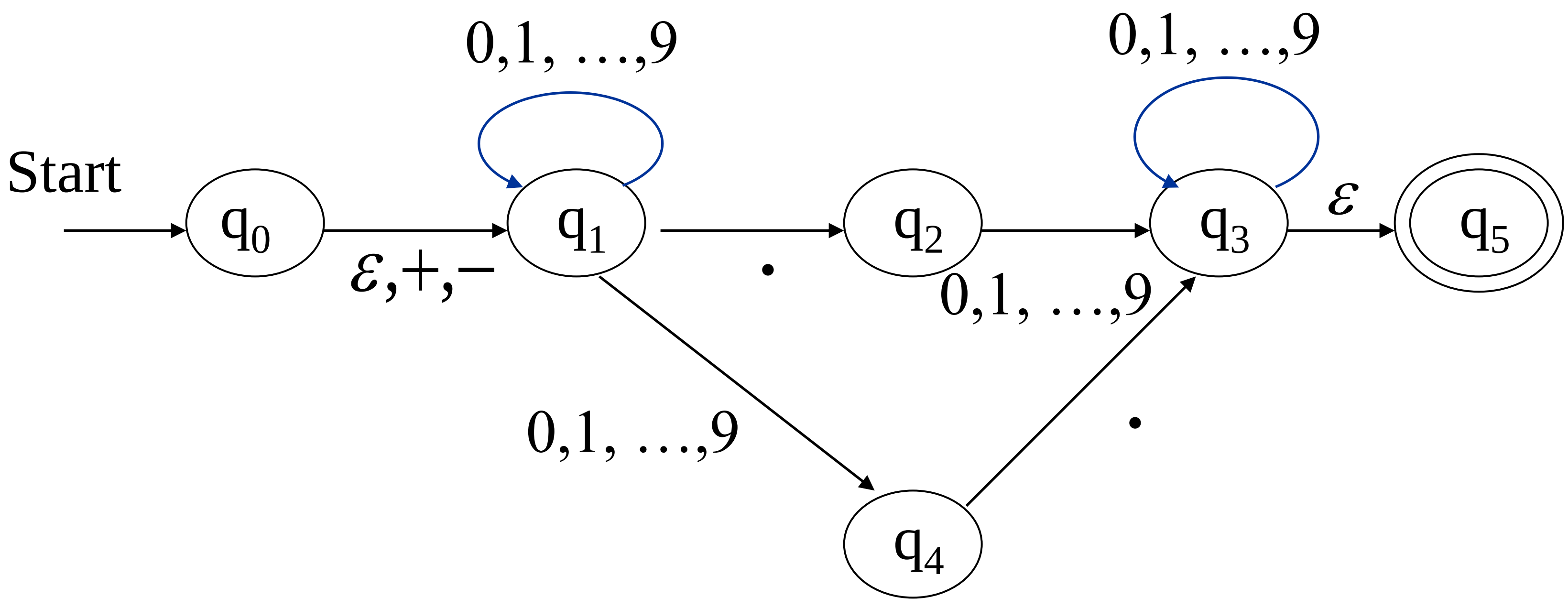
Example4. 1

Describe the language accepted by this NFA :



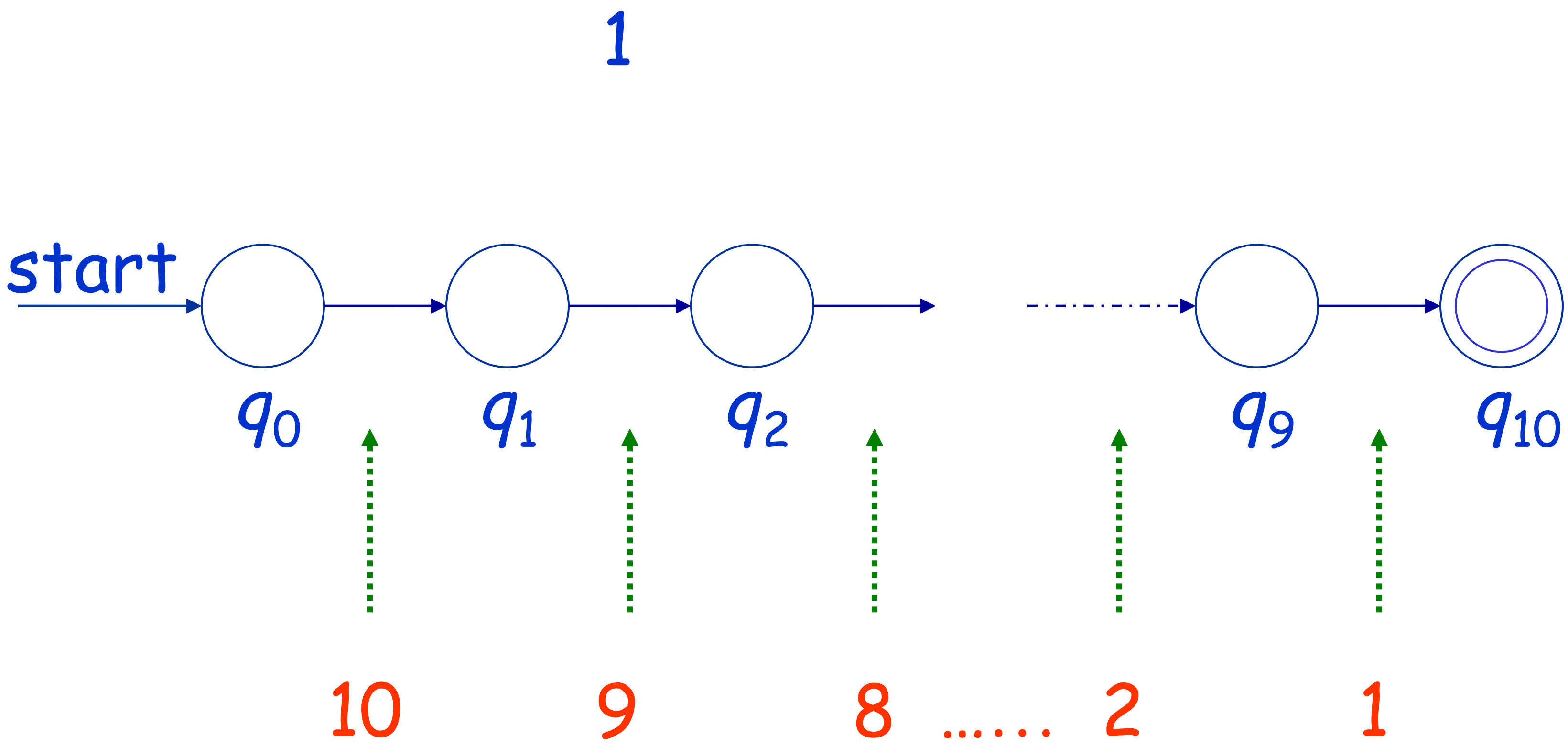
What about the NFA just accept decimal numbers ?

An ε- NFA for decimal numbers

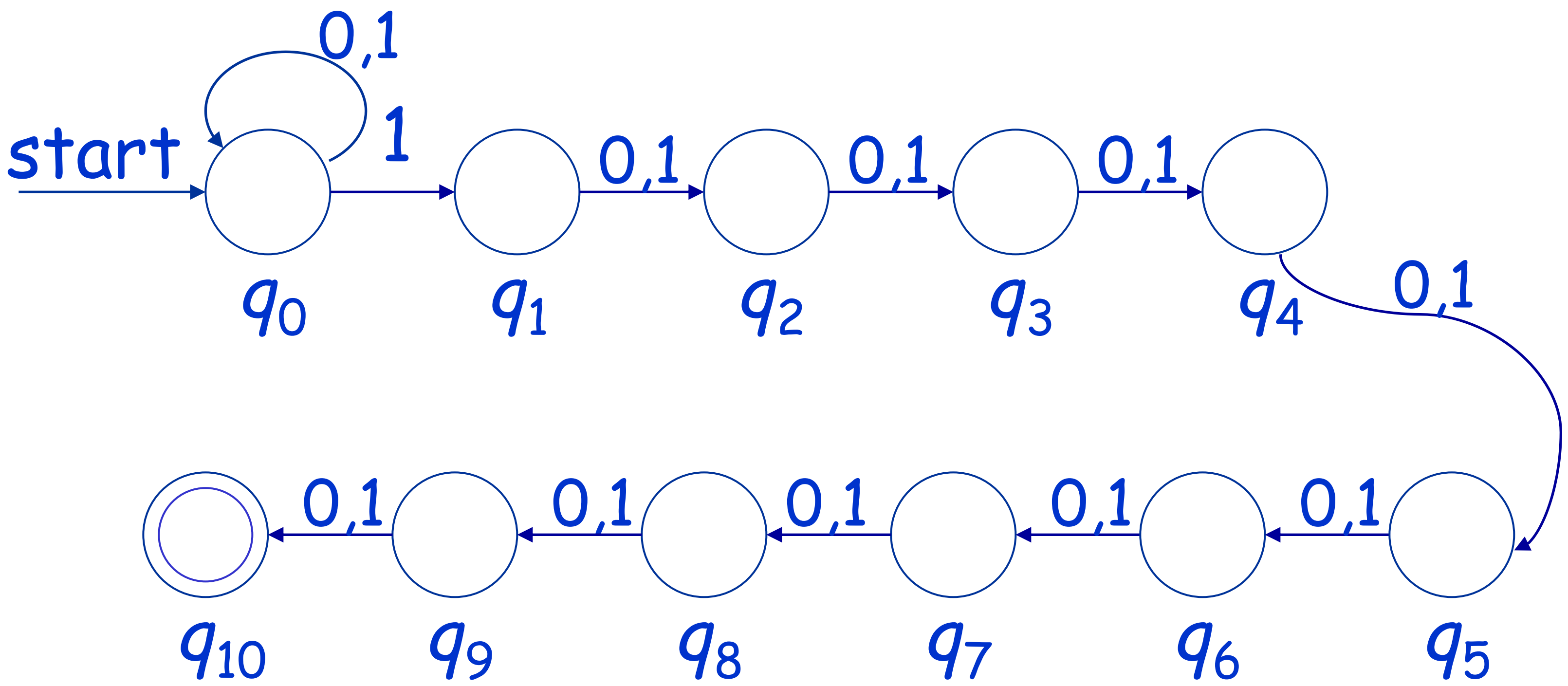


Example 4.2 Design an ε-NFA for following language

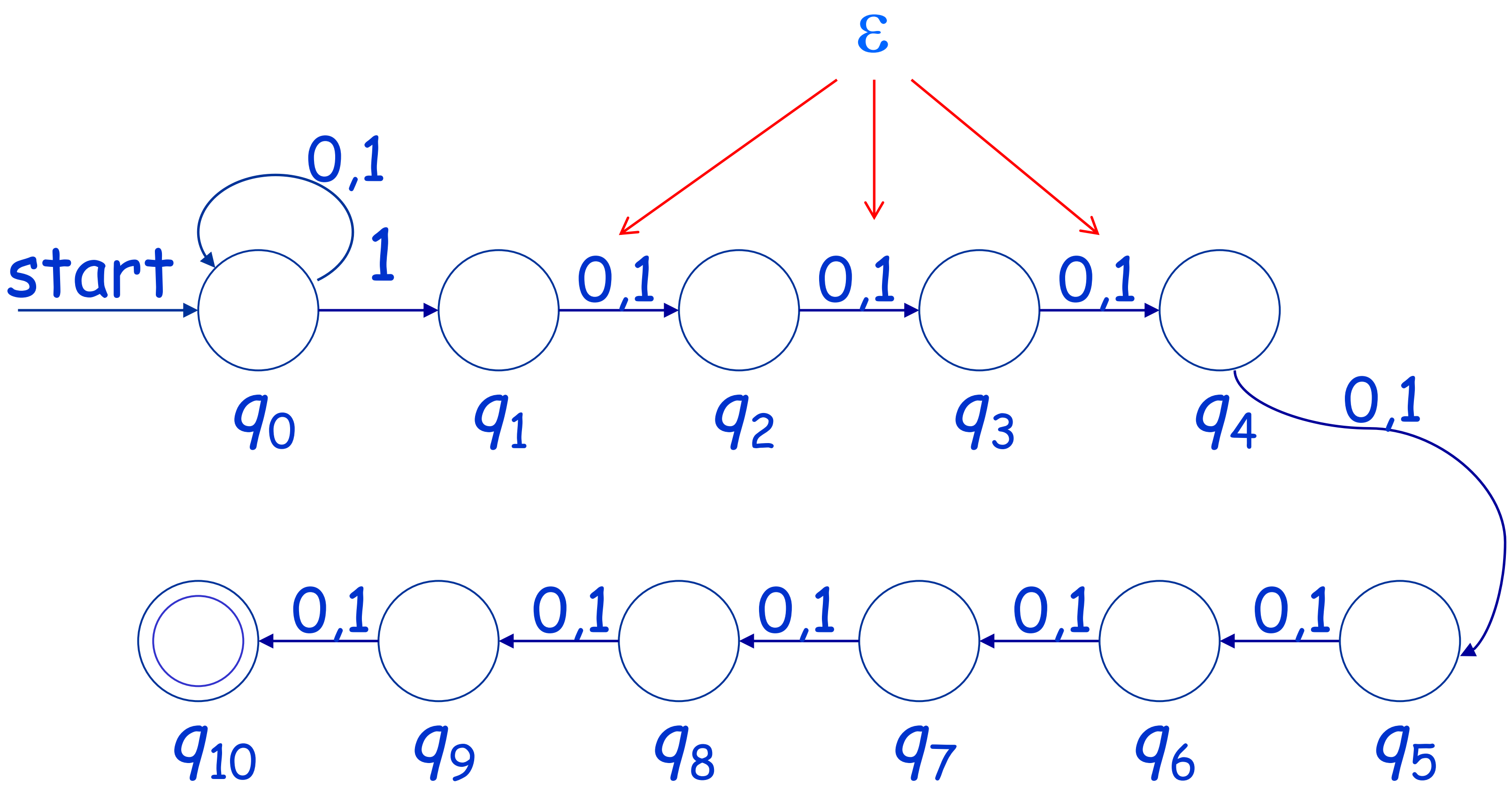
The set of strings of 0's and 1's such that at least one of the last ten positions is a 1 .



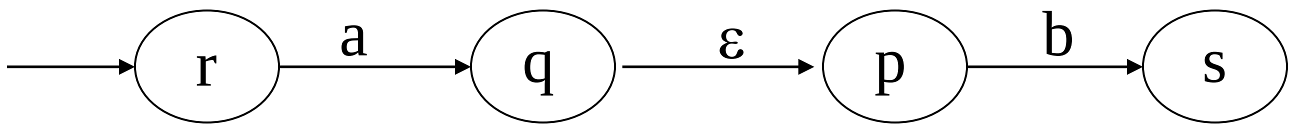
How about this NFA



How about this ε-NFA



ε- transition

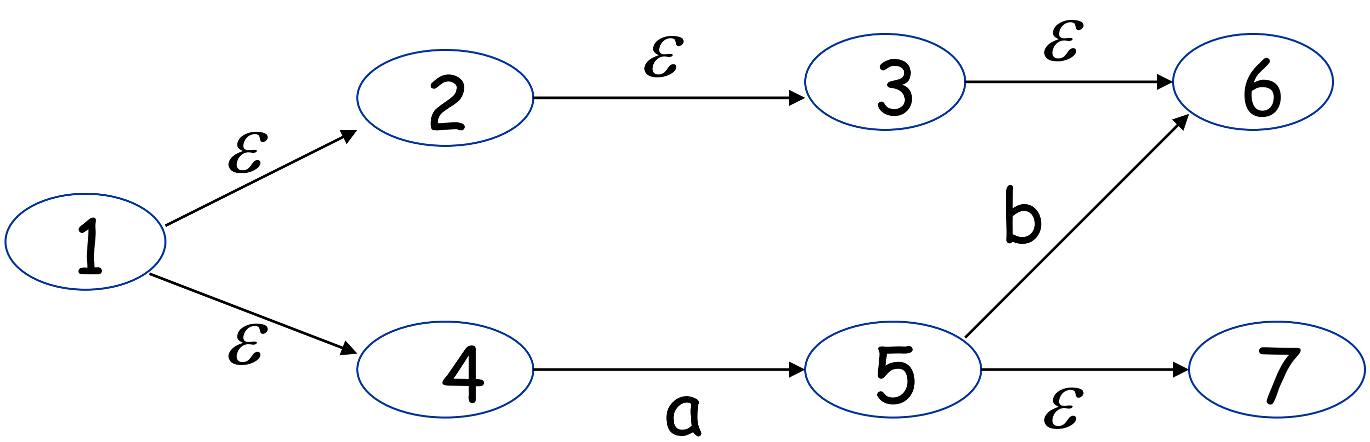


$\delta(r, a) = ?$ $\qquad$ $\delta(q, b) = ?$

ε- closure

BASIS : State *q* is in ECLOSE(*q*)

INDUCTION : If state *p* is in ECLOSE(*q*),and there is a transition from state *p* to state *r* labeled ε ,then *r* is in ECLOSE(*q*).



Extending transition to strings

BASIS :

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

INDUCTION :

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

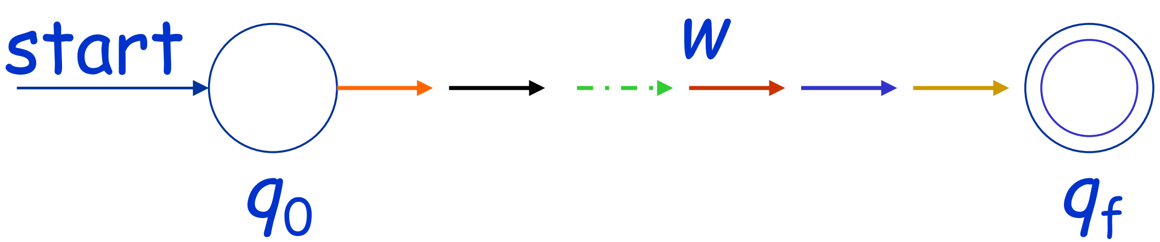
Let $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \cdots, r_m\}$

Then $\hat{\delta}(q,w) = \bigcup_{i=1}^{m} Eclose(r_i)$

The language of ε-NFA

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

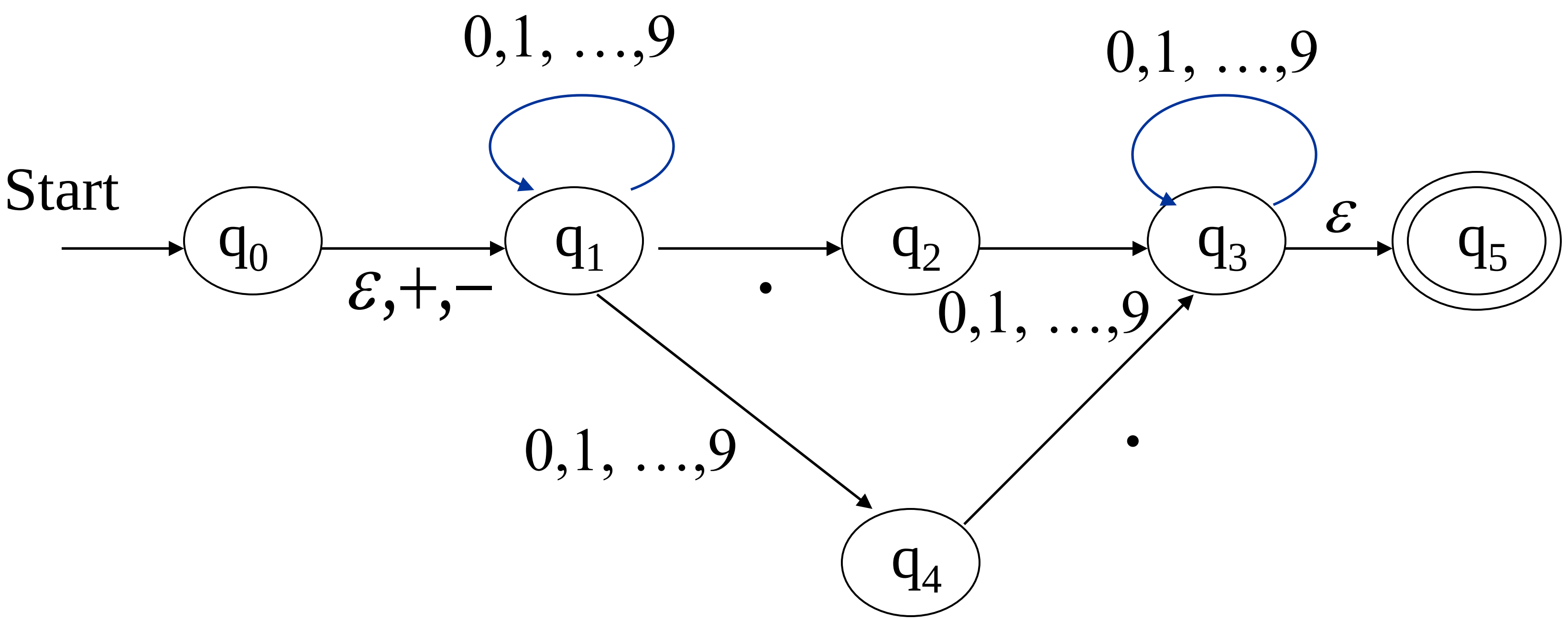
$$L(A) = \{w \mid \hat{\delta}(q_0, w) \cap F \neq \phi\}$$



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

Example 4.3

Compute : $\hat{\delta}(q_0, 5.6)$



Equivalence of states

- *equivalent states*

$$\forall w \in \Sigma^*, \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F$$

- *notice*

We never mentioned $\hat{\delta}(p, w)$ = $\hat{\delta}(q, w)$

- *distinguishable states*

$$\exists w \in \Sigma^*, \hat{\delta}(p, w) \in F \Leftrightarrow \neg\hat{\delta}(q, w) \in F$$

Example 6.5 Determine the equivalent states

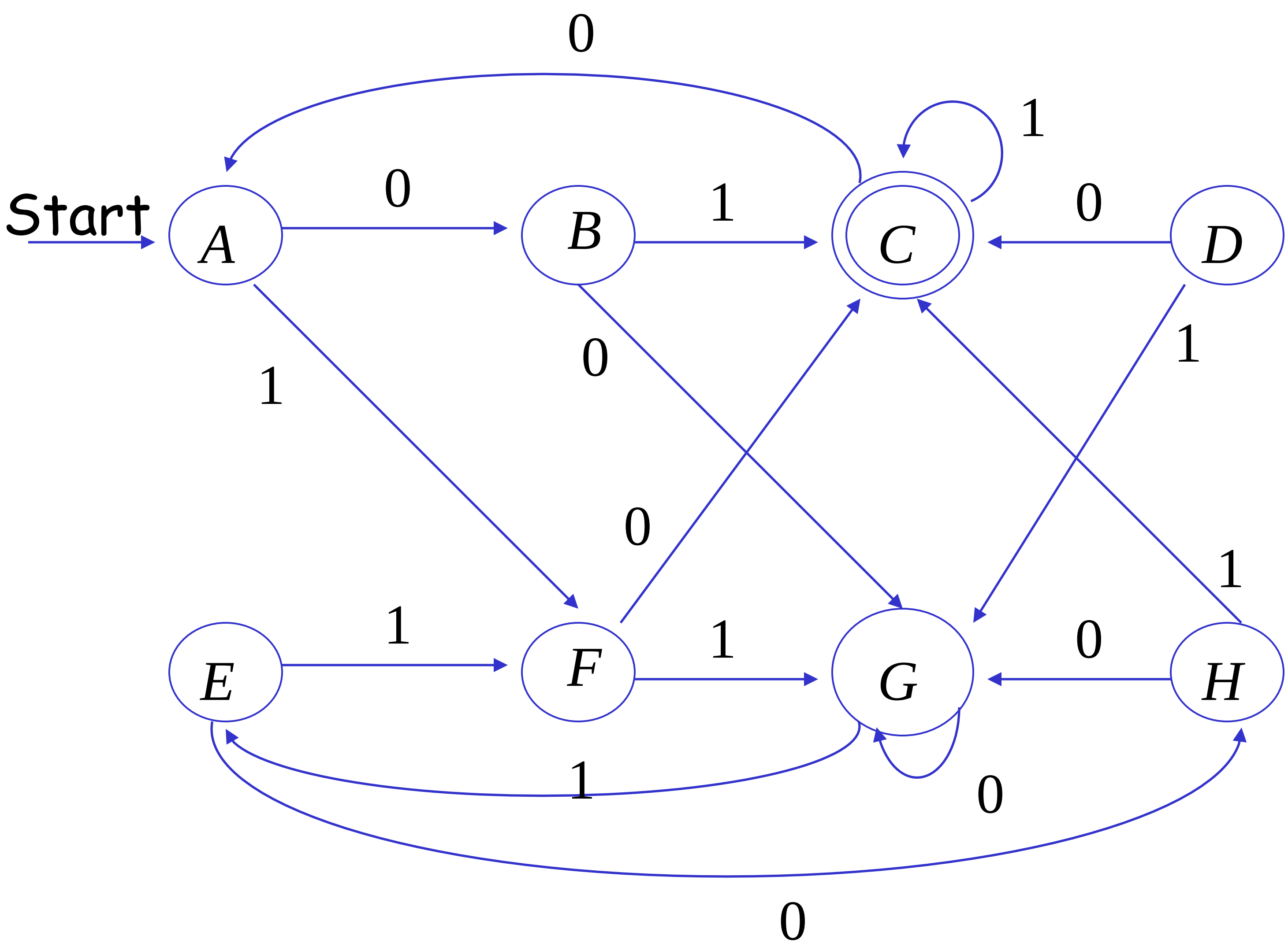


Table-filling algorithm

- *Basis* If p is accepting and q is not accepting, then p and q are distinguishable.
- *Induction* Let $r = \delta(p, a)$, $s = \delta(q, a)$, r and s are distinguishable. Then p and q are distinguishable.
- *Example*

B	×						
C	×	×					
D	×	×	×				
E		×	×				
F	×	×	×		×		
G	×	×	×	×	×	×	
H	×		×	×	×	×	×
	A	B	C	D	E	F	G

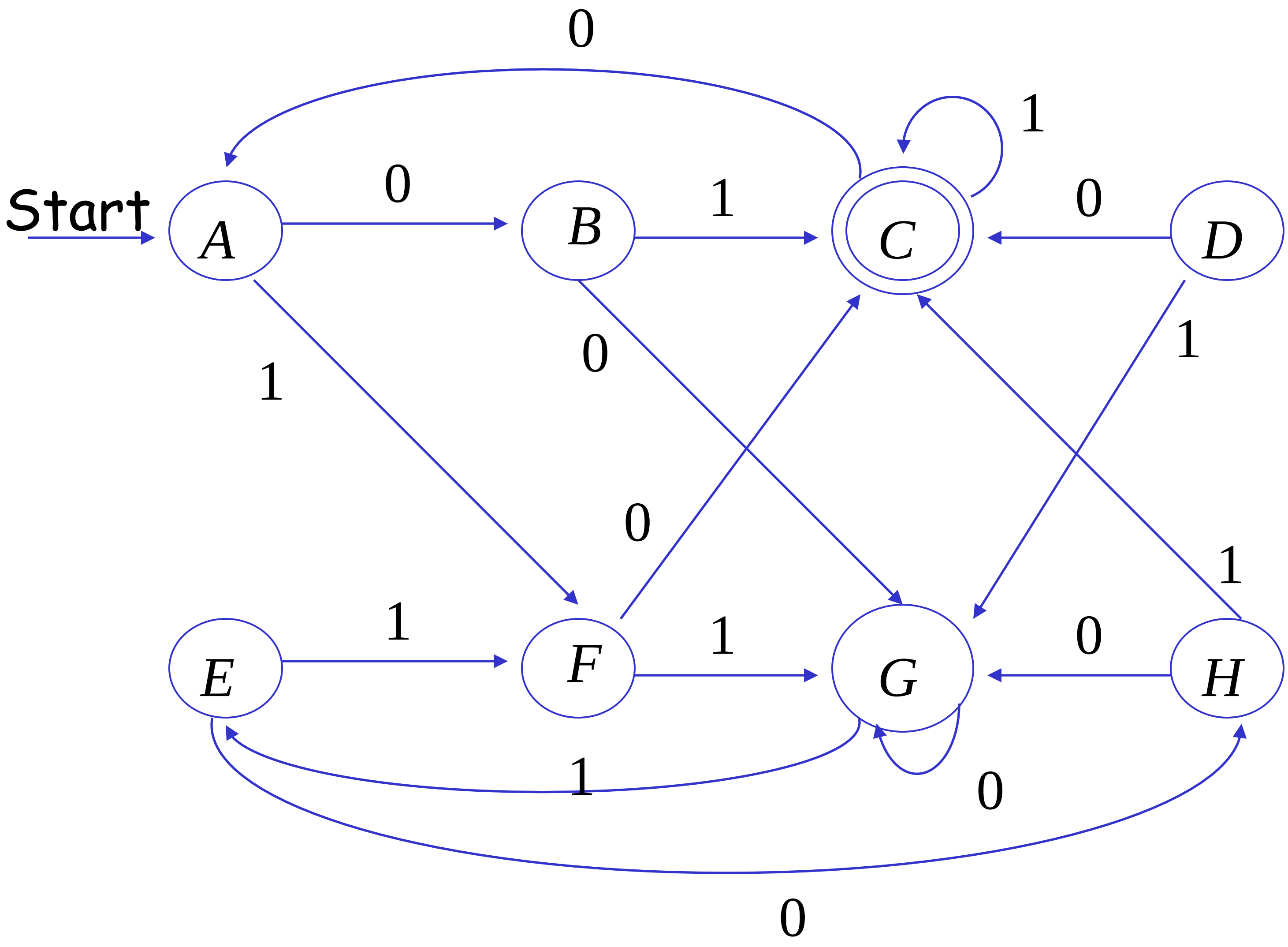
Minimization of DFA's

- what is minimization of DFA

- algorithm for minimization
  - partition remaining states into equivalent blocks
  - take blocks as states

- minimum-state DFA for a regular language is unique

Example 6.6 Minimization of DFA's



Summary

We have talked about three kinds of FA's

DFA -- NFA -- ε—NFA

- formal definition ⇒ $M = (Q, \Sigma, \delta, q_0, F)$

  DFA : $Q \times \Sigma \Rightarrow Q$, $\delta(q, a) = p$

  NFA : $Q \times \Sigma \Rightarrow 2^Q$, $\delta(q, a) = S$

  ε-NFA : $Q \times (\Sigma \cup \{\varepsilon\}) \Rightarrow 2^Q$, $\delta(q, a) = S$

- notation ⇒ formal + diagram + table
- construction ⇒ partitions of strings↔states

Summary

- language

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

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

- equivalence of DFA, NFA and ε-NFA

subset construction : (ε-) NFA $\Rightarrow$ DFA

- regular language $\Rightarrow$ { L | L=$L(A)$ and A is an FA }

Good good study

day day up !