第 四 讲

✧ 有限状态自动机

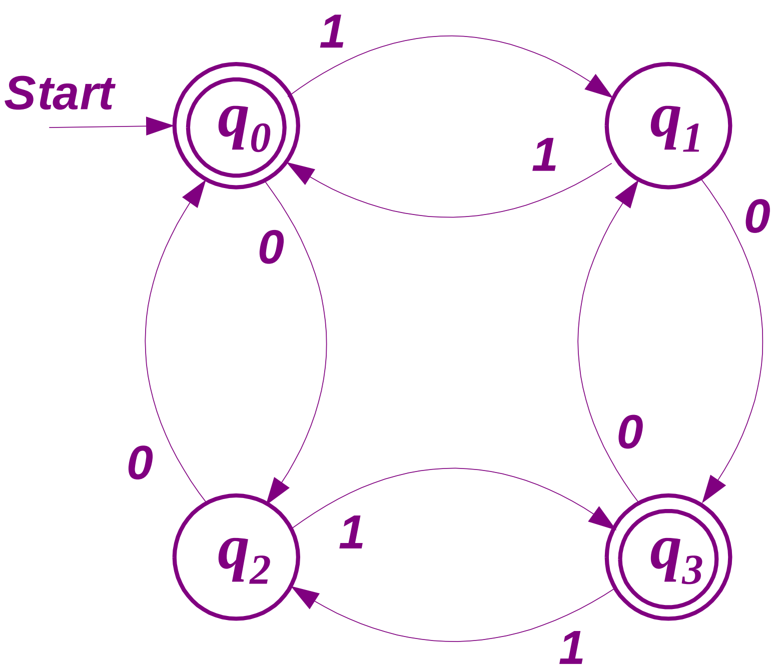
有限状态自动机

- ✧ 确定有限自动机
- ✧ 非确定有限自动机
- ✧ 确定与非确定有限自动机的等价性
- ✧ 有限自动机的一个应用 — 文本搜索
- ✧ 带 ε-转移的非确定有限自动机
- ✧ （确定）有限自动机的最小化

确定有限自动机

✧ 有限自动机的五要素

- 有限状态集
- 有限输入符号集
- 转移函数
- 一个开始状态
- 一个终态集合



确定有限自动机

✧ 确定有限自动机的形式定义

一个确定有限状态自动机 ***DFA (deterministic finite automata)*** 是一个五元组 $A = (Q, \Sigma, \delta, q_0, F)$.

– 有限状态集 Q

– 有限输入符号集 Σ

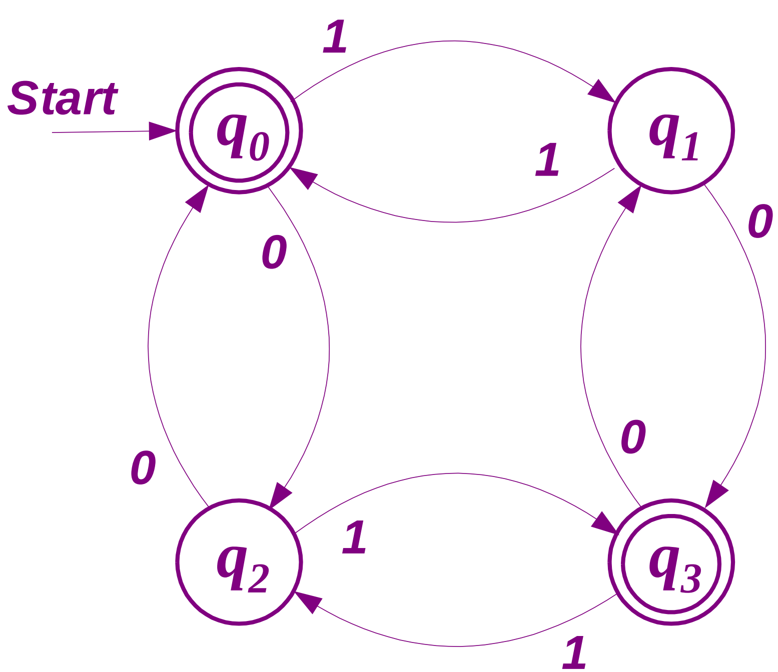
– 转移函数 δ: $\delta: Q \times \Sigma \to Q$

– 一个开始状态 q_0: $q_0 \in Q$

– 一个终态集合 F: $F \subseteq Q$

确定有限自动机

✧ 转移图表示的 DFA



- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $\delta(q_0, 0) = q_2$, $\delta(q_0, 1) = q_1$
  $\delta(q_1, 0) = q_3$, $\delta(q_1, 1) = q_0$
  $\delta(q_2, 0) = q_0$, $\delta(q_2, 1) = q_3$
  $\delta(q_3, 0) = q_1$, $\delta(q_3, 1) = q_2$
- q_0
- $F = \{q_0, q_3\}$

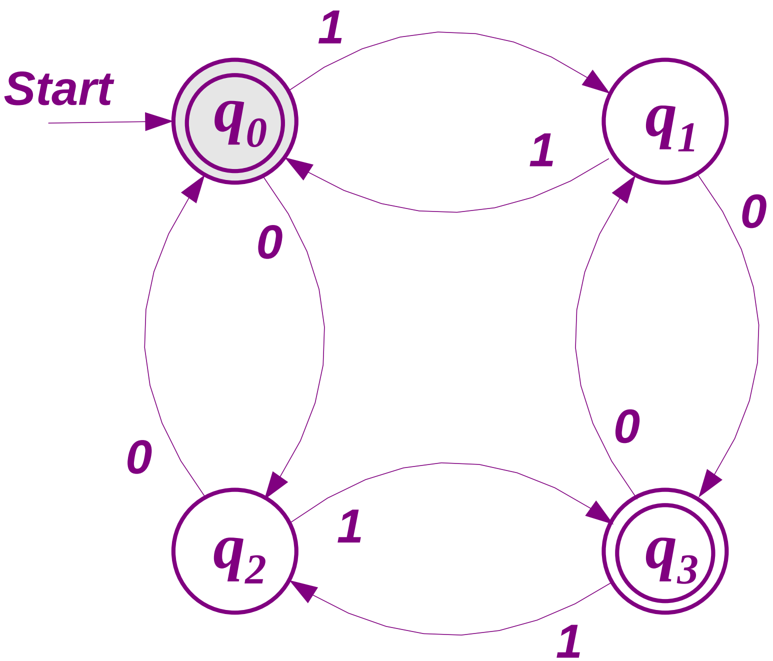
✧ 转移表表示的 DFA

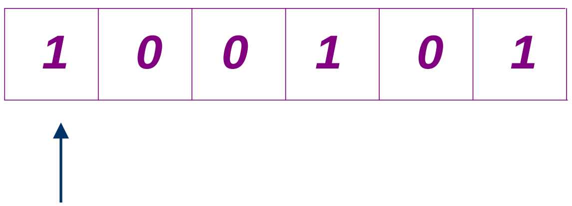
	0	1
$\rightarrow *q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
$*q_3$	q_1	q_2

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $\delta(q_0, 0) = q_2$, $\delta(q_0, 1) = q_1$
  $\delta(q_1, 0) = q_3$, $\delta(q_1, 1) = q_0$
  $\delta(q_2, 0) = q_0$, $\delta(q_2, 1) = q_3$
  $\delta(q_3, 0) = q_1$, $\delta(q_3, 1) = q_2$
- q_0
- $F = \{q_0, q_3\}$

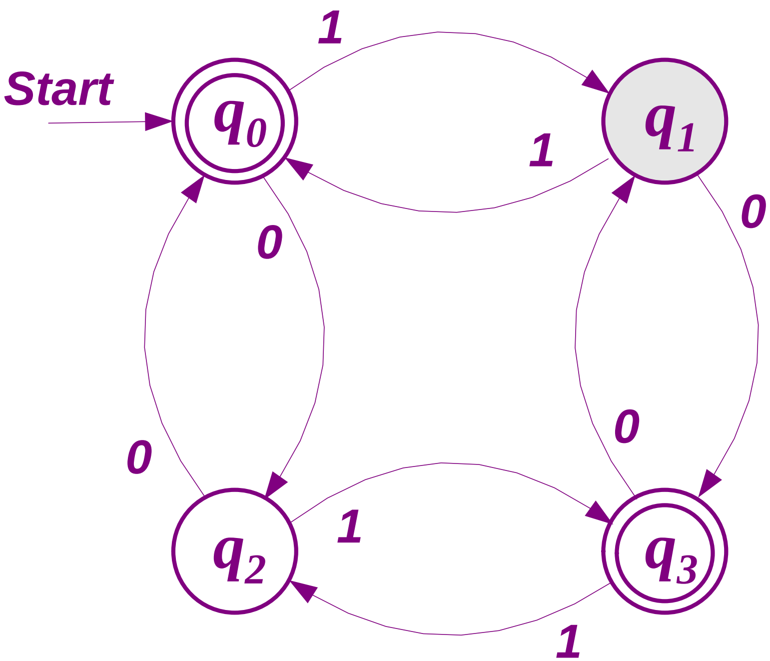
确定有限自动机

✧ DFA如何接受输入符号串



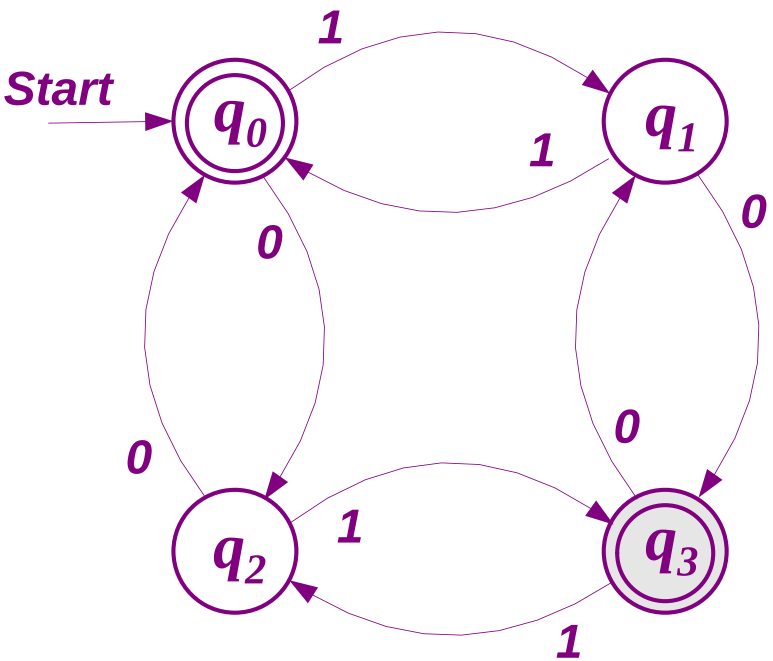


✧ DFA如何接受输入符号串



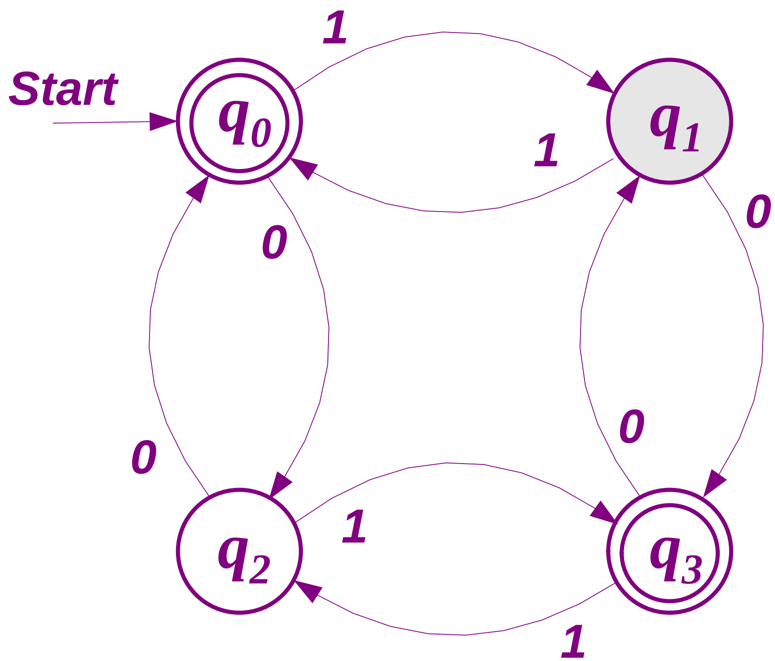
1	0	0	1	0	1

✧ DFA如何接受输入符号串



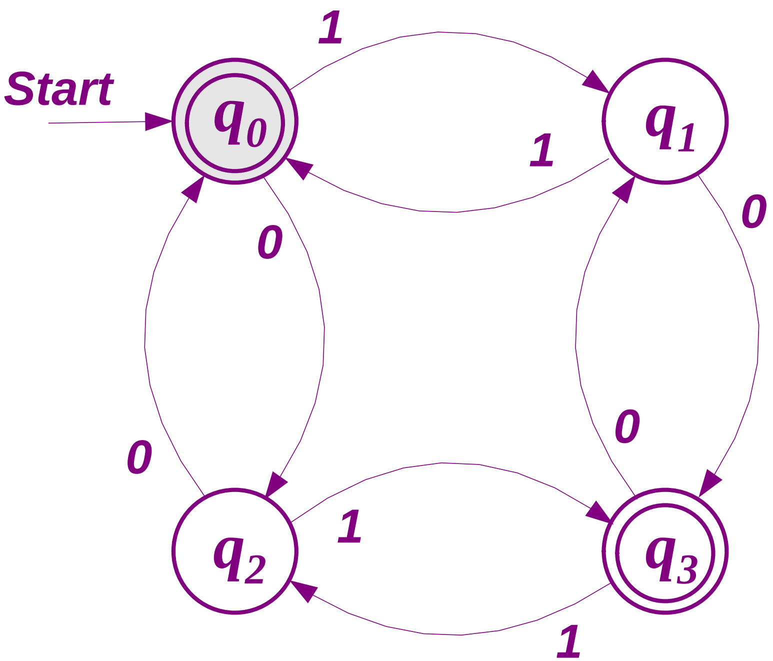
1	0	0	1	0	1

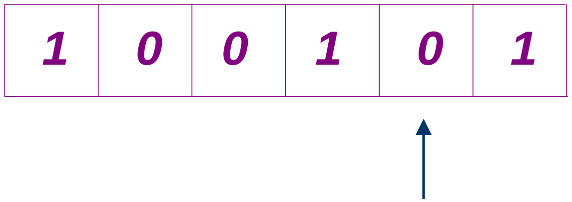
- DFA如何接受输入符号串



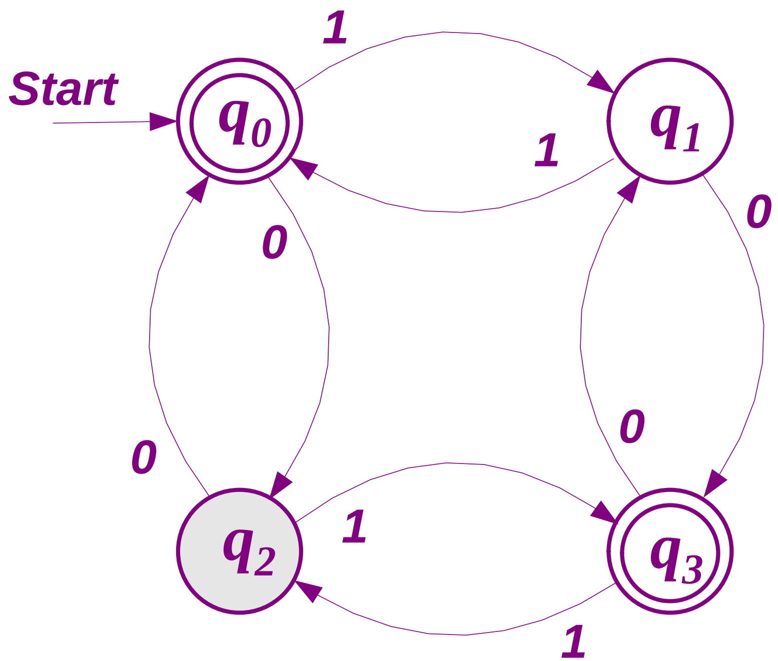
1	0	0	1	0	1

✧ DFA如何接受输入符号串



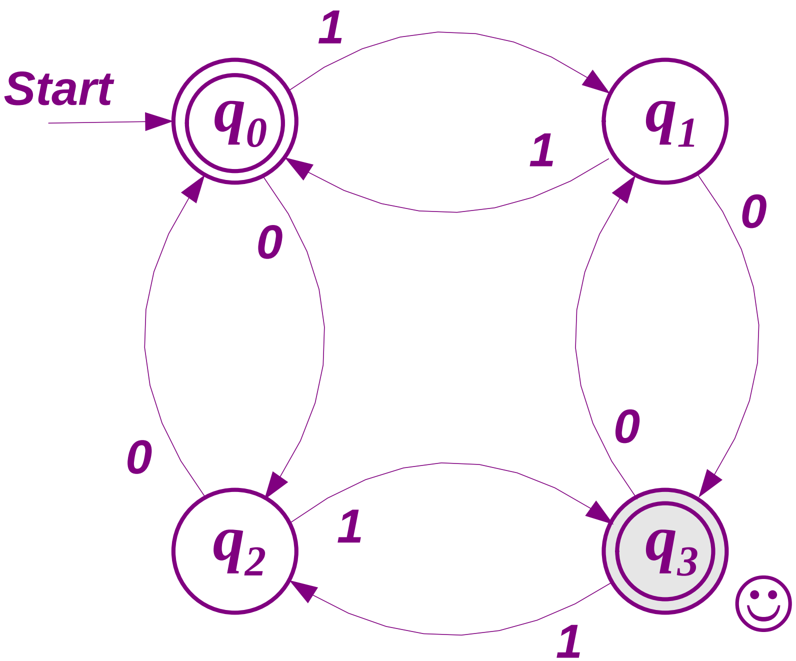


✧ DFA如何接受输入符号串



1	0	0	1	0	1

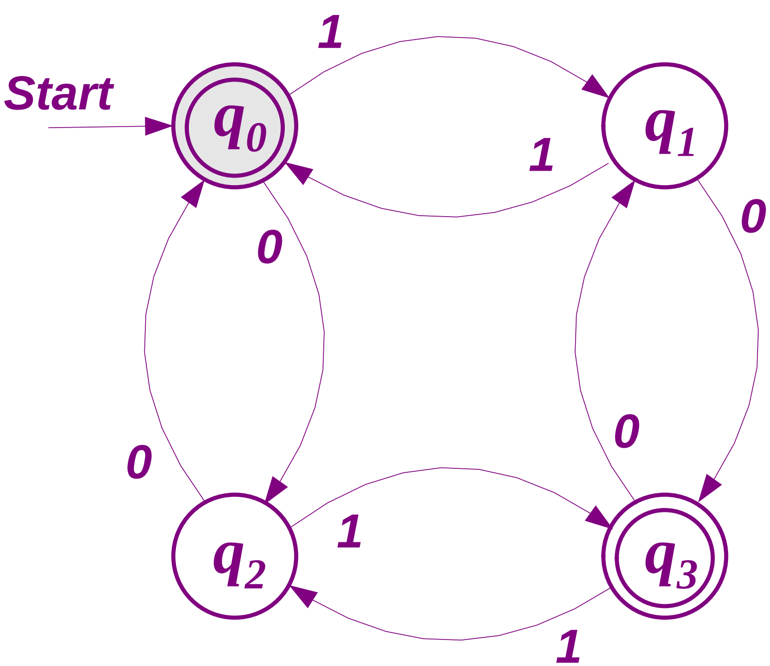
✧ DFA如何接受输入符号串

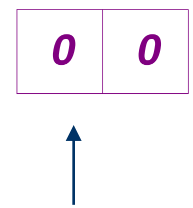


1	0	0	1	0	1

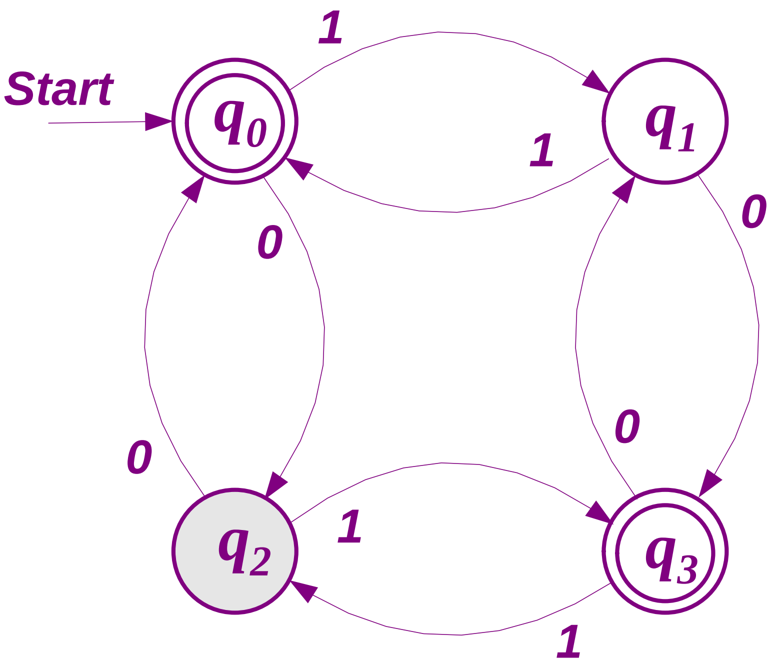
确定有限自动机

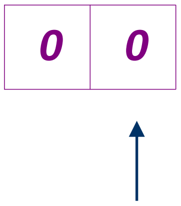
✧ DFA如何接受输入符号串



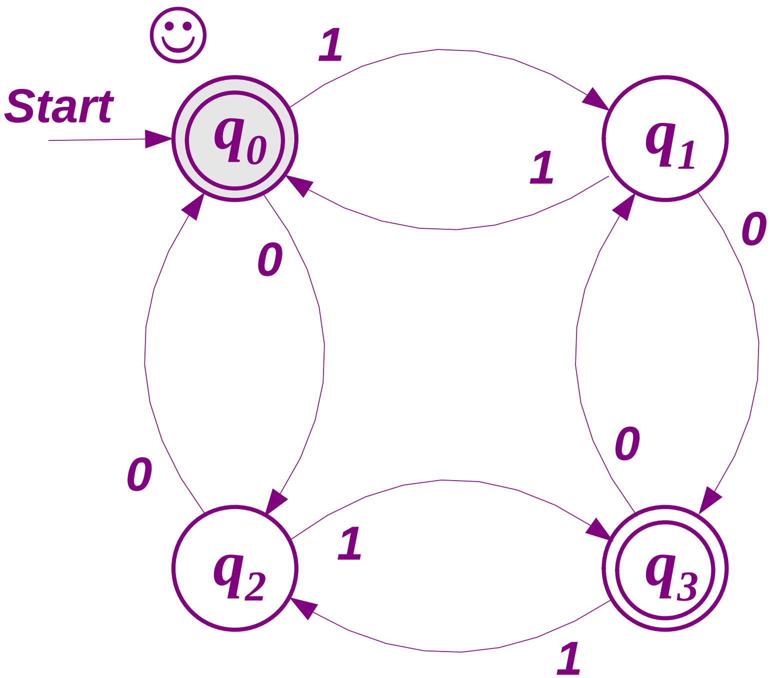


✧ DFA如何接受输入符号串



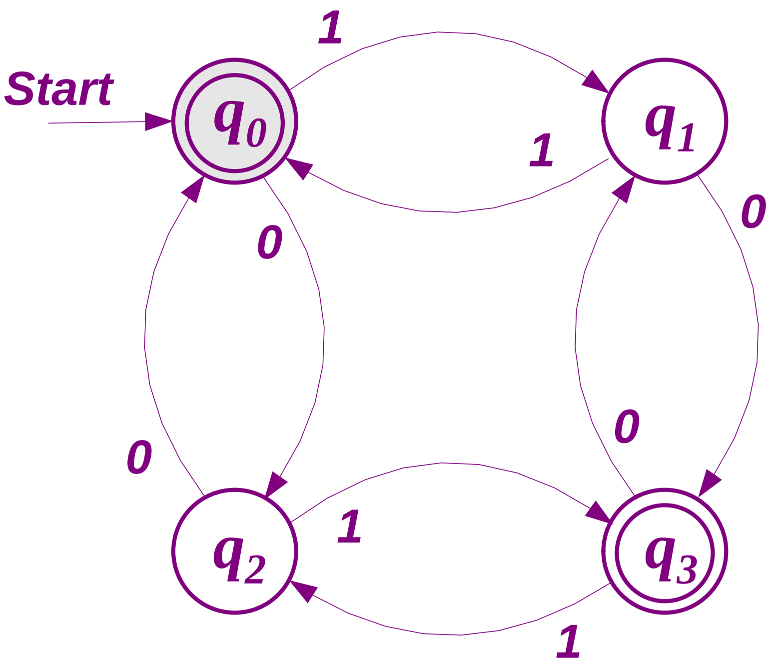


✧ DFA如何接受输入符号串



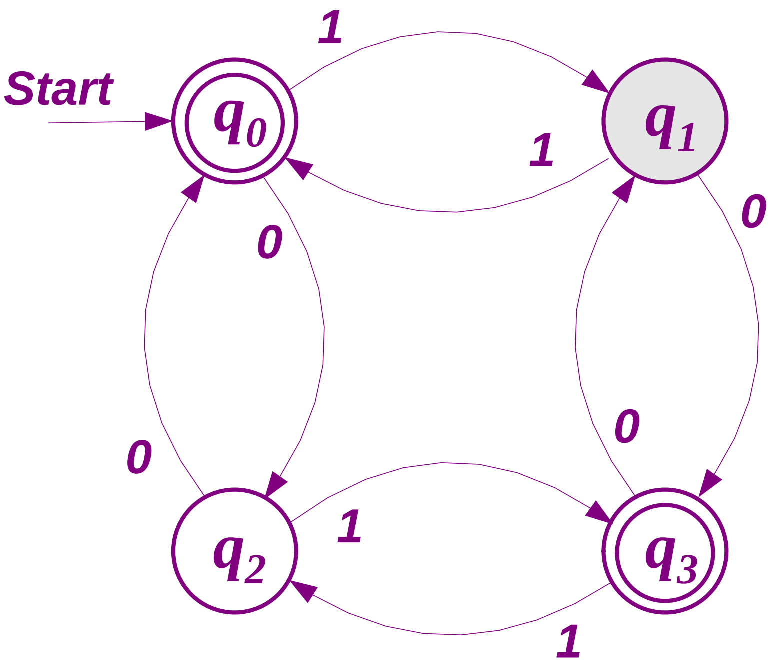
0	0

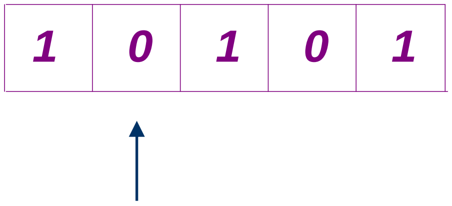
✧ DFA如何接受输入符号串



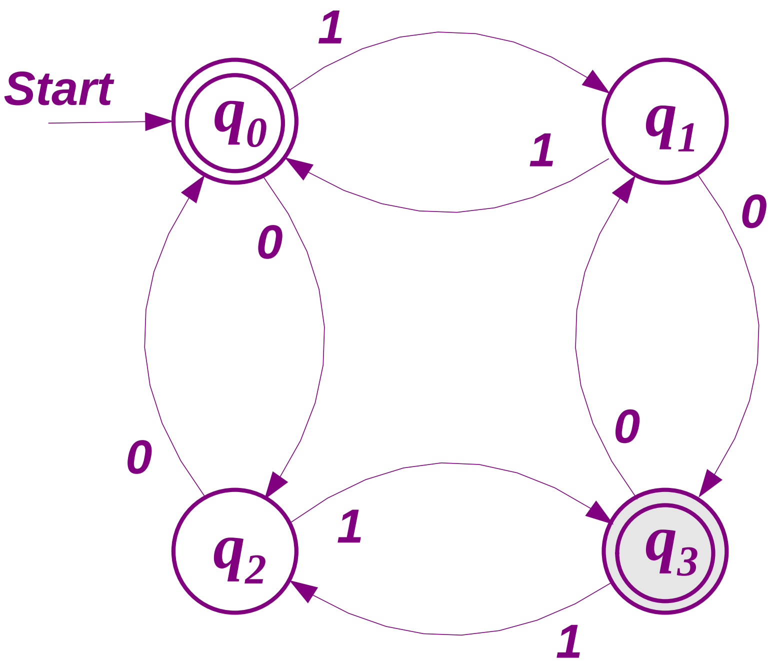
1	0	1	0	1

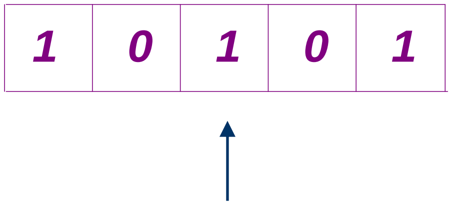
✧ DFA如何接受输入符号串



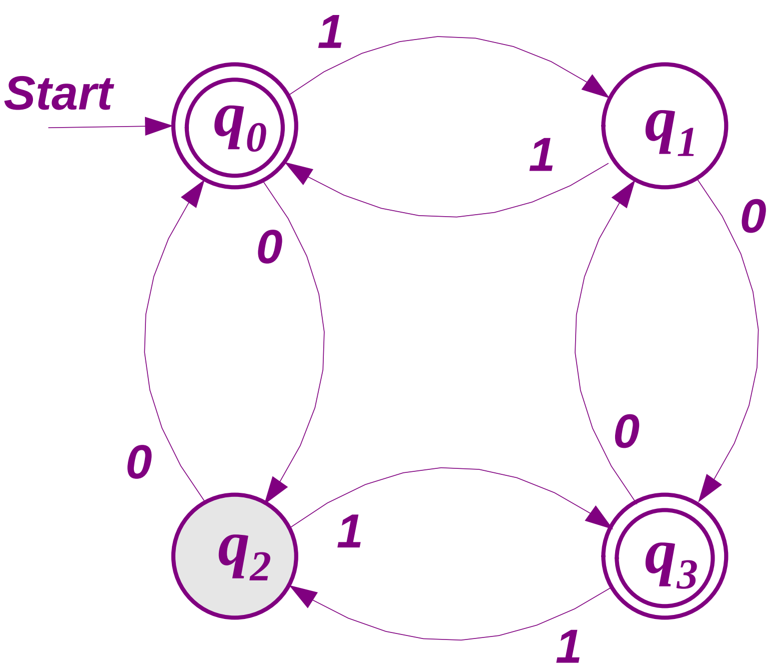


✧ DFA如何接受输入符号串



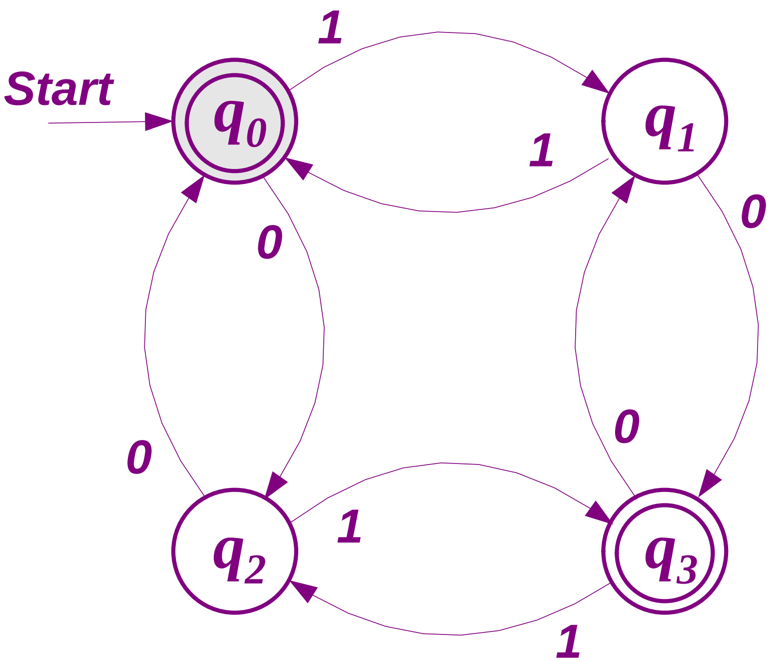


✧ DFA如何接受输入符号串



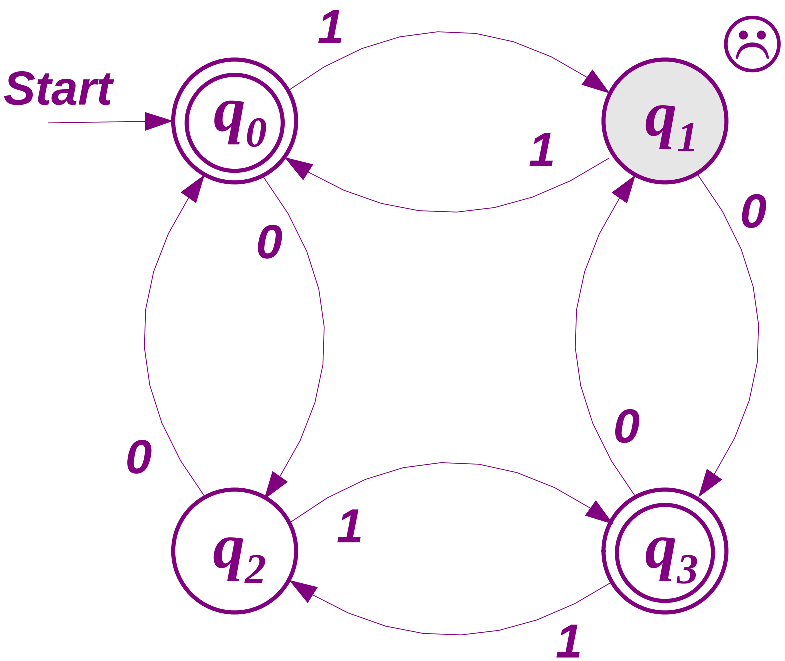
1	0	1	0	1

✧ DFA如何接受输入符号串



1	0	1	0	1

✧ DFA如何接受输入符号串



1	0	1	0	1

确定有限自动机

✧ 扩展转移函数适合于输入字符串

– 设一个 DFA $A = (Q, \Sigma, \delta, q_0, F)$

$\delta: Q \times \Sigma \to Q$

– 扩充定义 $\delta': Q \times \Sigma^* \to Q$

对任何$q \in Q$，定义:

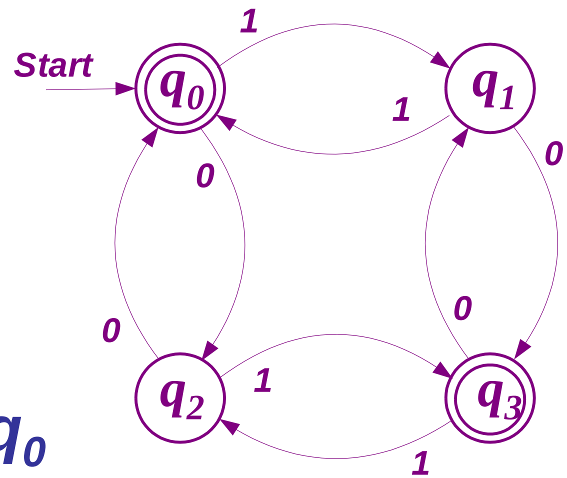
1 $\delta'(q, \varepsilon) = q$

2 若$w = xa$, 其中 $x \in \Sigma^*$, $a \in \Sigma$, 则

$$\delta'(q, w) = \delta(\delta'(q, x), a)$$

✧ 扩展转移函数适合于输入字符串

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



– 举例

$$\delta'(q_0, \varepsilon) = q_0$$

$$\delta'(q_0, 0) = \delta(q_0, 0) = q_2$$

$$\delta'(q_0, 00) = \delta(q_2, 0) = q_0$$

$$\delta'(q_0, 001) = \delta(q_0, 1) = q_1$$

$$\delta'(q_0, 0010) = \delta(q_1, 0) = q_3$$

✧ DFA 的语言

– 设一个 DFA $A = (Q, \Sigma, \delta, q_0, F)$

– 定义 A 的语言:

$$L(A) = \{ w \mid w \in \Sigma^* \wedge \delta'(q_0, w) \in F \}$$

– 设 L 是 Σ 上的语言，如果存在一个 DFA $A = (Q, \Sigma, \delta, q_0, F)$，满足 $L = L(A)$ ，则可以证明 L 是一个正规语言.

✧ DFA 的语言

– 举例　Σ = { 0, 1 } 上的语言 L = { w | w 中 0 、1 数目的奇偶性相同 }，则 L 是一个正规语言. 可证 L是如下DFA 的语言.

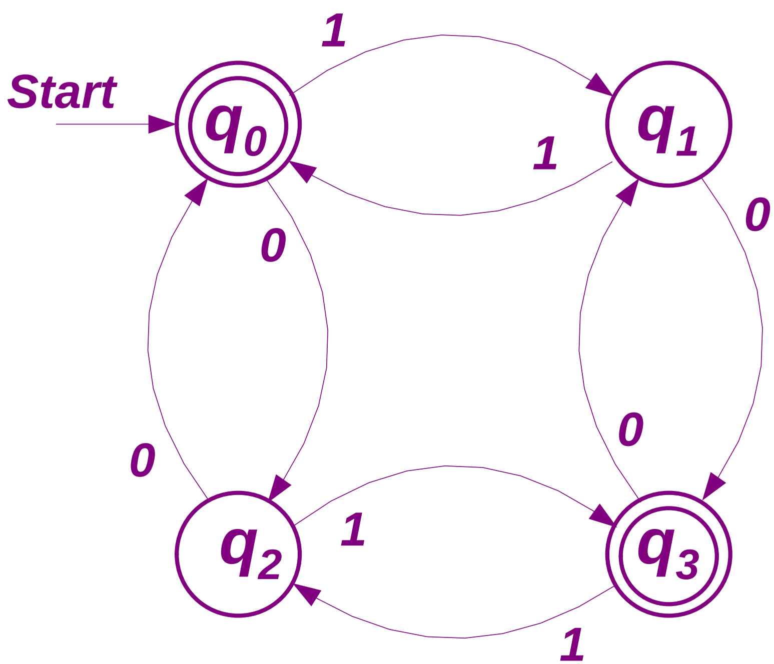
– 证明

留作思考题

(采用互归纳法,

参考 Example 2.4

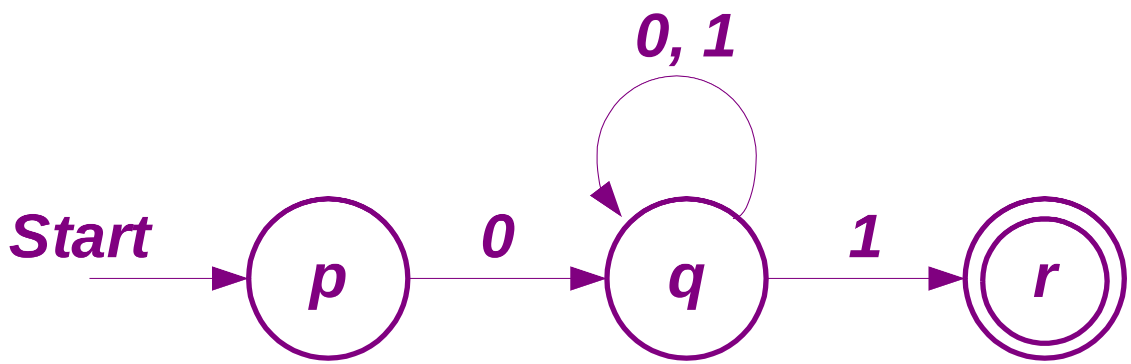
和 Example 1.23)



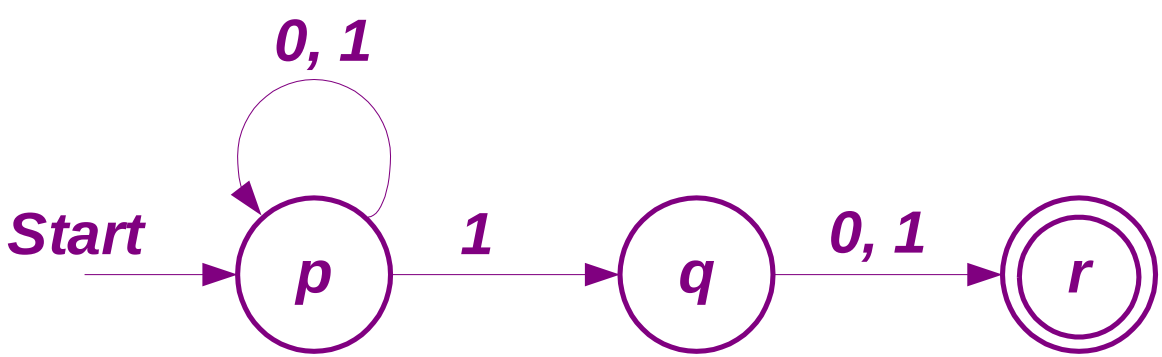
非确定有限自动机

✧ 非确定有限自动机举例

(1)



(2)



非确定有限自动机

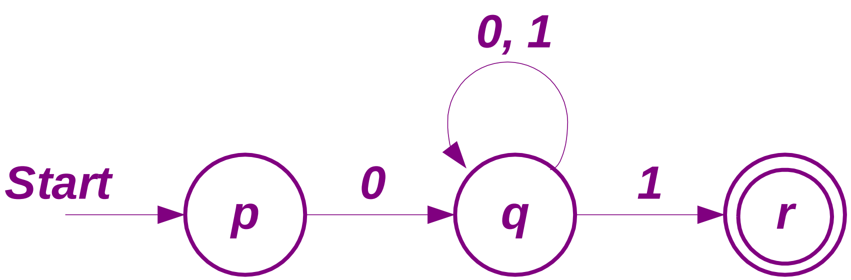
✧ 非确定有限自动机的形式定义

一个非确定有限状态自动机 ***NFA nondeterministic finite automata)*** 是一个五元组 $A = (Q, \Sigma, \delta, q_0, F)$.

- 有限状态集 — Q
- 有限输入符号集 — Σ
- 转移函数 — δ
- 一个开始状态 — q_0　　$q_0 \in Q$
- 一个终态集合 — F　　$F \subseteq Q$
- 与 ***DFA*** 唯一不同之处　$\delta: Q \times \Sigma \rightarrow 2^Q$

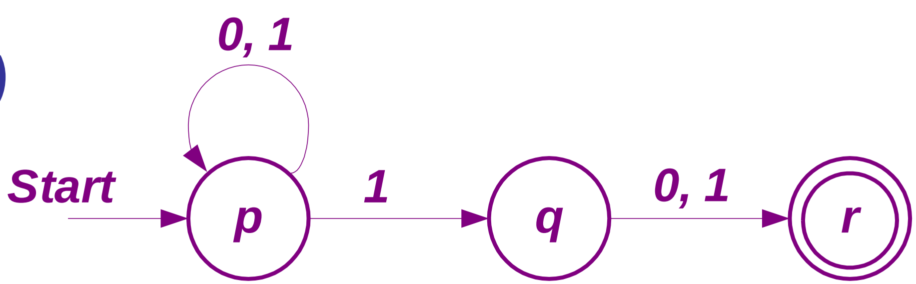
✧ 转移图和转移表表示的NFA

(1)



	0	1
$\rightarrow p$	$\{q\}$	ϕ
q	$\{q\}$	$\{q, r\}$
$* r$	ϕ	ϕ

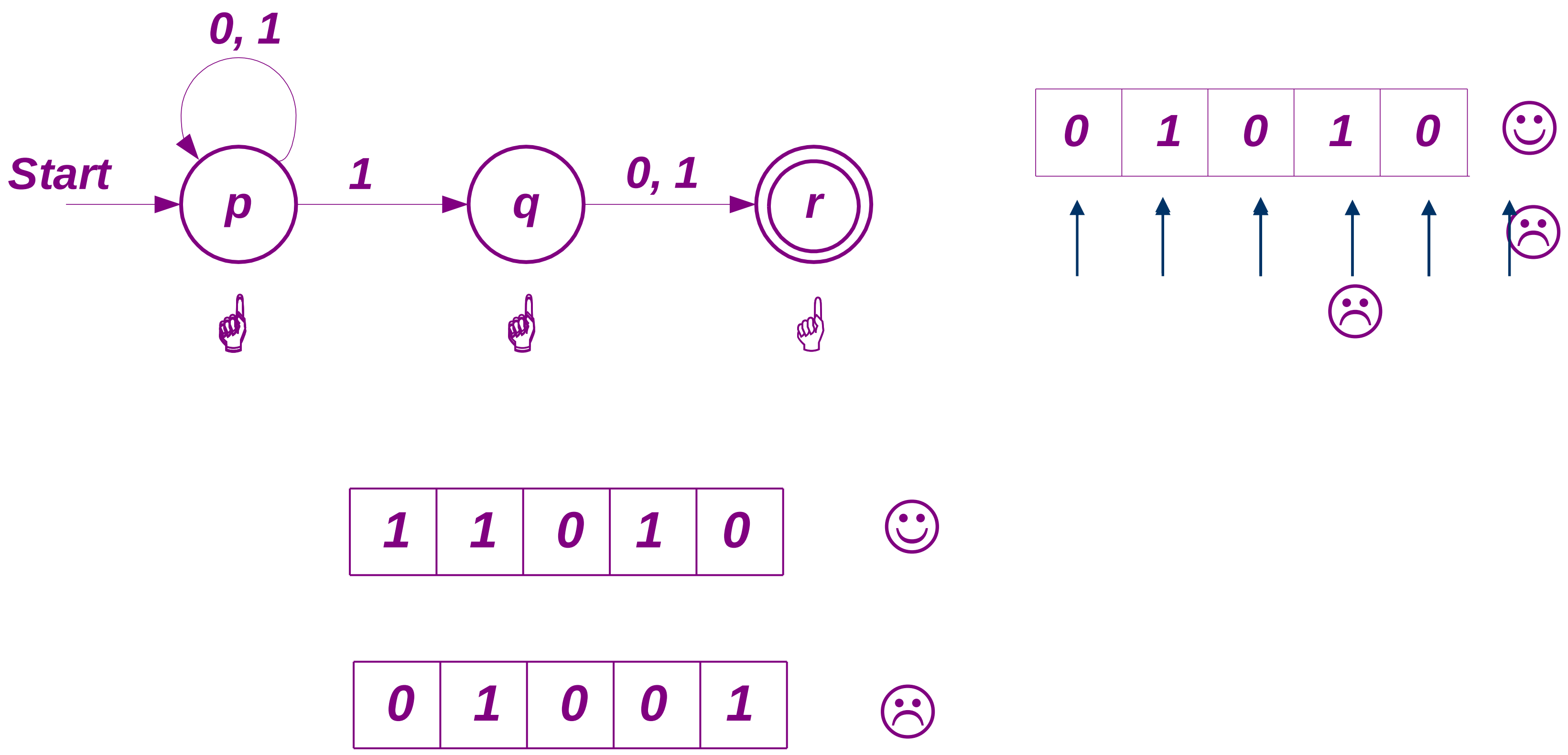
(2)



	0	1
$\rightarrow p$	$\{p\}$	$\{p, q\}$
q	$\{r\}$	$\{r\}$
$* r$	ϕ	ϕ

非确定有限自动机

✧ NFA 如何接受输入符号串



非确定有限自动机

✧ 扩展转移函数适合于输入字符串

- 设一个 NFA $A = (Q, \Sigma, \delta, q_0, F)$
- $\delta: Q \times \Sigma \to 2^Q$
- 扩充定义 $\delta': Q \times \Sigma^* \to 2^Q$
- 对任何$q \in Q$，定义：

1 $\delta'(q, \varepsilon) = \{ q \}$

2 若$w = xa$, 其中 $x \in \Sigma^*$, $a \in \Sigma$, 并且假设

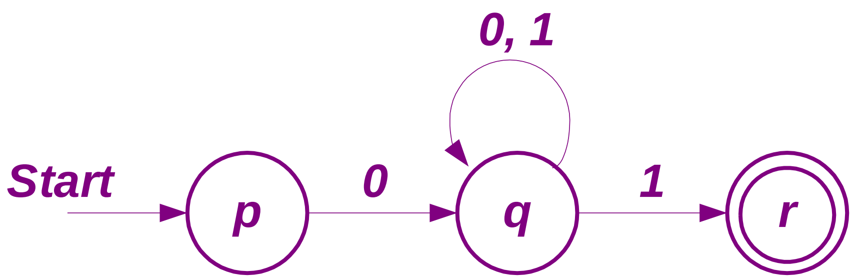
$\delta'(q, x) = \{ p_1, p_2, \ldots, p_k \}$, 则

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

非确定有限自动机

✧ 扩展转移函数适合于输入字符串

	0	1
$\rightarrow p$	$\{q\}$	ϕ
q	$\{q\}$	$\{q, r\}$
$* r$	ϕ	ϕ



– 举例

$\delta'(p, \varepsilon) = \{p\}$

$\delta'(p, 0) = \{q\}$

$\delta'(p, 01) = \{q, r\}$

$\delta'(p, 010) = \{q\}$

$\delta'(p, 0100) = \{q\}$

$\delta'(p, 01001) = \{q, r\}$

✧ NFA 的语言

- 设一个 NFA $A = (Q, \Sigma, \delta, q_0, F)$
- 定义 A 的语言：

$$L(A) = \{ w \mid w \in \Sigma^* \wedge \delta'(q_0, w) \cap F \neq \phi \}$$

- 设 L 是 Σ 上的语言，如果存在一个 NFA $A = (Q, \Sigma, \delta, q_0, F)$，满足 $L = L(A)$，则可以证明 L 也是一个正规语言.

DFA 和 NFA 的等价性

定理: *L* 是某个 *DFA* 的语言, 当且仅当 *L* 也是某个 *NFA* 的语言.

证明: 分两步证明.

(1) 设 *L* 是某个 *DFA D* 的语言, 则存在一个 *NFA N*, 满足 $L(N) = L(D) = L$;

(2) 设 *L* 是某个 *NFA N* 的语言, 则存在一个 *DFA D*, 满足 $L(D) = L(N) = L$;

DFA 和 NFA 的等价性

✧ 从 *DFA* 构造等价的 *NFA*

– 设 L 是某个 DFA $D = (Q, \Sigma, \delta_D, q_0, F)$ 的语言, 则存在一个 NFA N, 满足 $L(N) = L(D) = L$.

– **证明:** 定义$N = (Q, \Sigma, \delta_N, q_0, F)$, 其中$\delta_N$定义为

- 对$q \in Q$ 和$a \in \Sigma$,
  若 $\delta_D(q,a) = p$, 则 $\delta_N(q,a) = \{p\}$.

需要证明: 对任何$w \in \Sigma^*$,

$$\delta'_D(q_0, w) = p \text{ iff } \delta'_N(q_0, w) = \{p\}.$$

归纳于 $|w|$ 易证上述命题.

DFA 和 NFA 的等价性

✧ 从 NFA 构造等价的 DFA（子集构造法）

– 设 L 是某个 NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ 的语言, 则存在一个 DFA D, 满足 $L(D) = L(N) = L$.

– 证明: 定义 $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$, 其中

- $Q_D = \{ S \mid S \subseteq Q_N \}$
- 对 $S \in Q_D$ 和 $a \in \Sigma$, $\delta_D(S, a) = \bigcup_{q \in S} \delta_N(q,a)$
- $F_D = \{S \mid S \subseteq Q_N \wedge S \cap F_N \neq \phi \}$

需要证明: 对任何 $w \in \Sigma^*$,

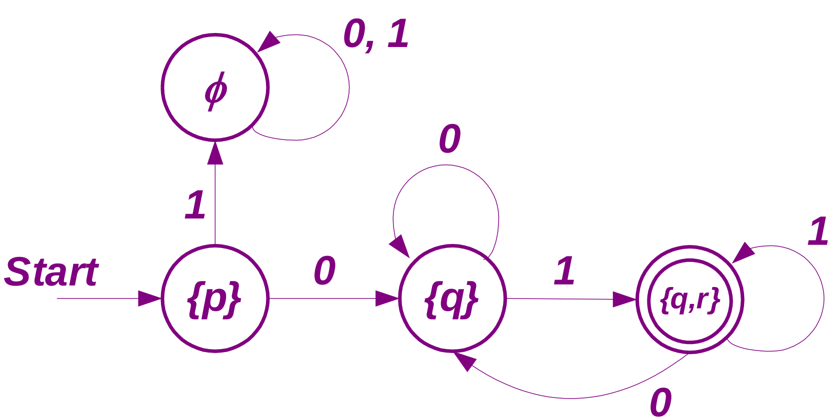
$$\delta'_D(\{q_0\}, w) = \delta'_N(q_0, w).$$

归纳于 $|w|$ 可证上述命题.

DFA 和 NFA 的等价性

✧ 子集构造法举例

	0	1
$\rightarrow p$	$\{q\}$	ϕ
q	$\{q\}$	$\{q, r\}$
$* r$	ϕ	ϕ



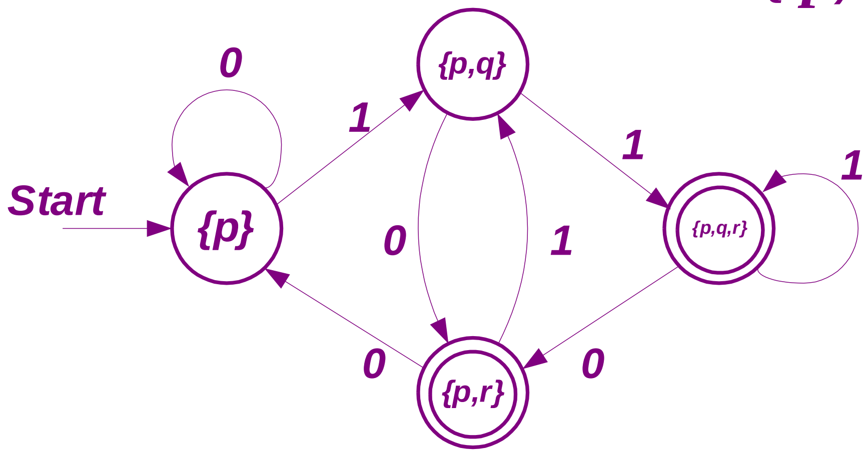
	0	1
ϕ	ϕ	ϕ
$\rightarrow \{p\}$	$\{q\}$	ϕ
$\{q\}$	$\{q\}$	$\{q, r\}$
$* \{r\}$	ϕ	ϕ
$\{p, q\}$	$\{q\}$	$\{q, r\}$
$* \{p, r\}$	$\{q\}$	ϕ
$* \{q, r\}$	$\{q\}$	$\{q, r\}$
$* \{p, q, r\}$	$\{q\}$	$\{q, r\}$

DFA 和 NFA 的等价性

✧ 子集构造法举例

	0	1
$\rightarrow p$	$\{p\}$	$\{p, q\}$
q	$\{r\}$	$\{r\}$
$* r$	ϕ	ϕ

	0	1
$\rightarrow \{p\}$	$\{p\}$	$\{p, q\}$
$\{p, q\}$	$\{p, r\}$	$\{p, q, r\}$
$*\{p, r\}$	$\{p\}$	$\{p, q\}$
$*\{p, q, r\}$	$\{p, r\}$	$\{p, q, r\}$



DFA 和 NFA 的等价性

✧ 从 NFA 构造等价的 DFA（子集构造法）

定理: 设 $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ 是一个 NFA，通过子集构造法得到相应的DFA $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$，则对任何$w \in \Sigma^*$，$\delta'_D(\{q_0\}, w) = \delta'_N(q_0, w)$.

证明: 归纳于 $|w|$

1 设$|w| = 0$，即 $w = \varepsilon$.

由定义知 $\delta'_D(\{q_0\}, \varepsilon) = \delta'_N(q_0, \varepsilon) = \{q_0\}$.

2 设$|w| = n+1$，并 $w = xa$, $a \in \Sigma$.　注意到$|x| = n$.

假设 $\delta'_D(\{q_0\}, x) = \delta'_N(q_0, x) = \{p_1, p_2, \ldots, p_k\}$.

则 $\delta'_D(\{q_0\}, w) = \delta_D(\delta'_D(\{q_0\}, x), a)$

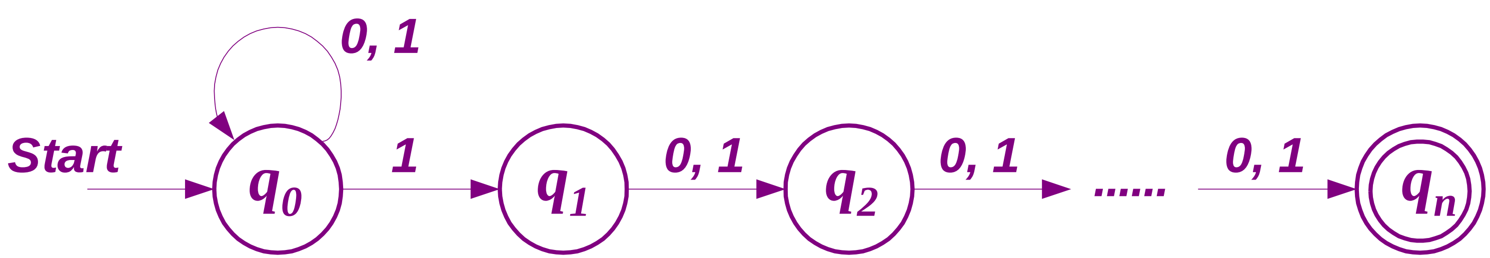
$$= \delta_D(\{p_1, p_2, \ldots, p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a).$$

$$= \delta'_N(q_0, w)$$

DFA 和 NFA 的等价性

✧ 子集构造法得到的状态数

- 实践中，通过子集构造法得到的 ***DFA*** 的状态数目与原 ***NFA*** 的状态数目大体相当
- 在较坏的情况下，上述 ***DFA*** 的状态数目接近于所有子集的数目
- 举例 由如下 ***NFA*** 构造的 ***DFA*** 的状态数目至少为 2^n



DFA 和 NFA 的等价性

✧ 子集构造法得到的状态数

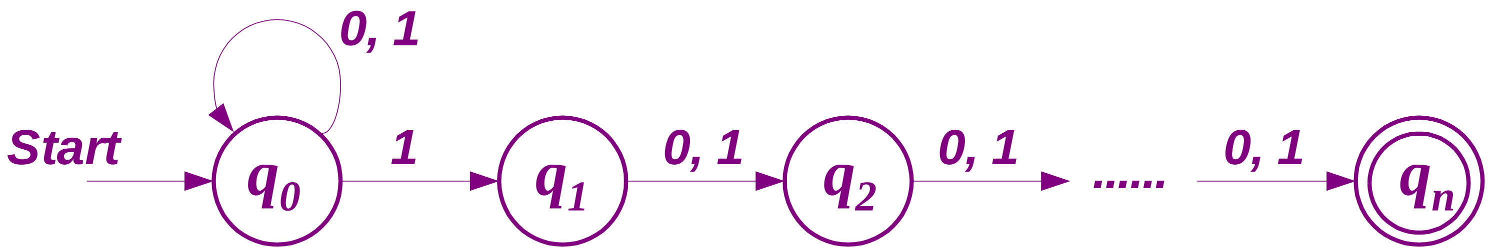
– 上页例子的证明要点，采用反证法

假设由此 NFA 构造的 DFA 的状态数目少于2^n

考虑长度为 n 的 0,1 串共有 2^n 个，所以存在两个不同的串$a_1a_2…a_n$和 $b_1b_2…b_n$ 做为该 DFA 的输入，可以到达同一状态 q.（by Pigeonhole Principle）

若$a_1 \neq b_1$，则 q 既是终态又是非终态，矛盾；

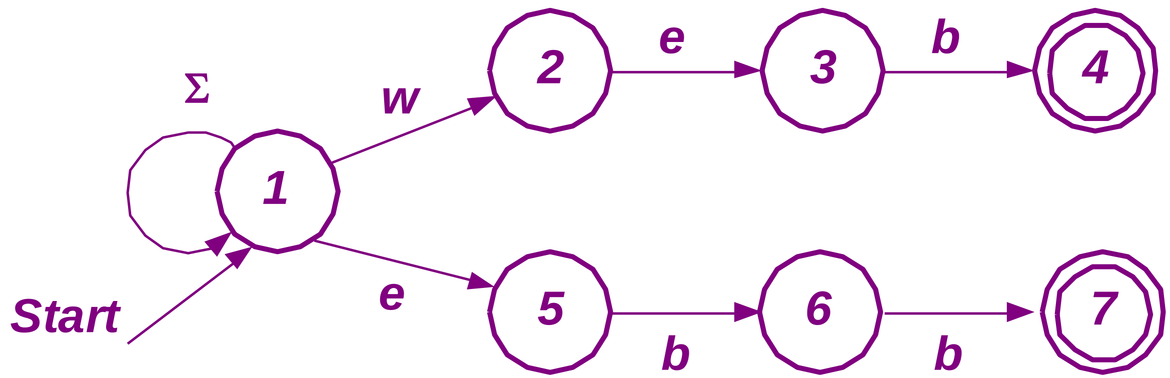
一般情况，若 $a_k \neq b_k$，设$a_1a_2…a_n00…0$（k-1 个 0）或$b_1b_2…b_n00…0$（k-1 个 0）作为输入串时该 DFA 到达状态p，则 p 既是终态又是非终态，矛盾。



文本搜索

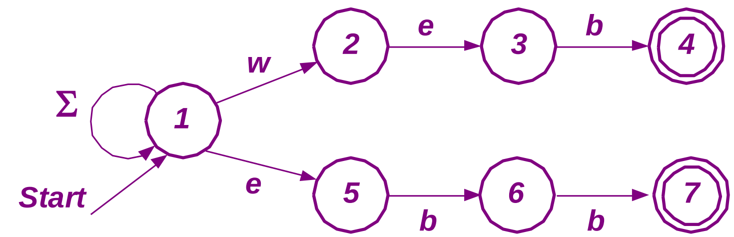
✧ 举例 设计一个 *NFA* 用来在文本中搜索字符串*web* 和*ebb.*

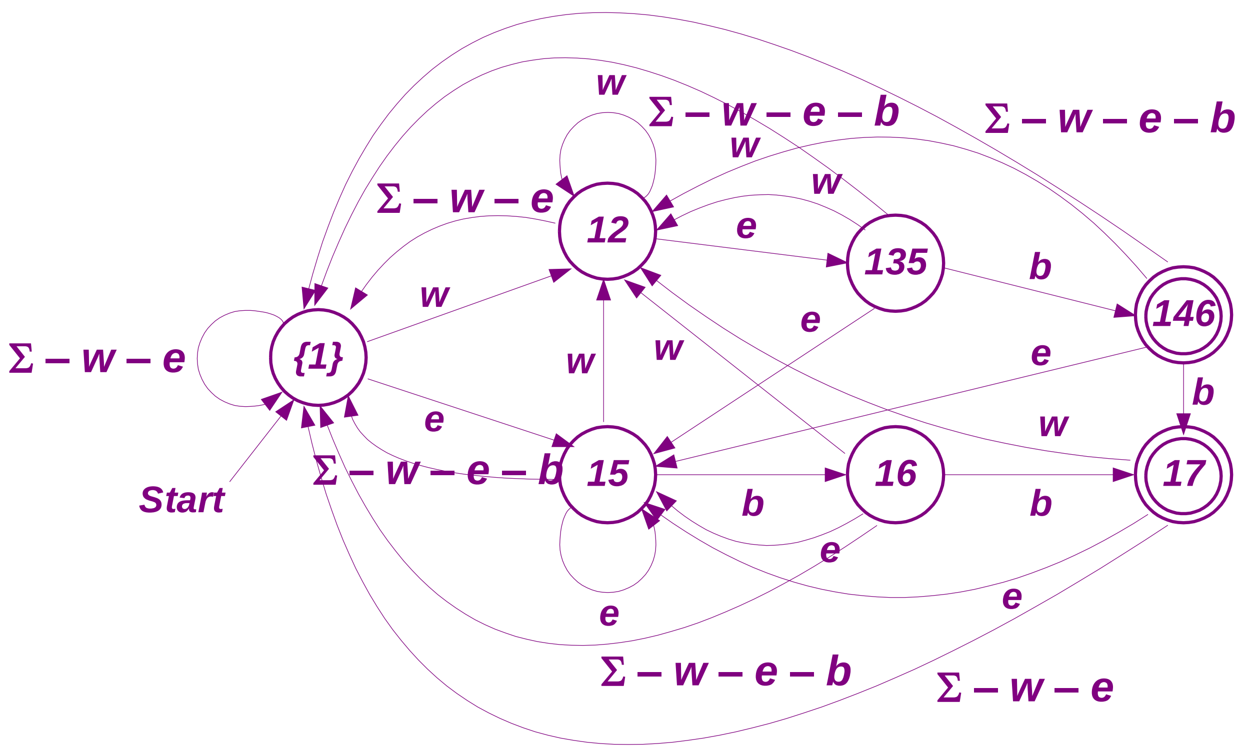
✧ 解 下图为一个满足条件的 *NFA*，其中 Σ 代表所有 *ASCII* 字符的集合.



文本搜索

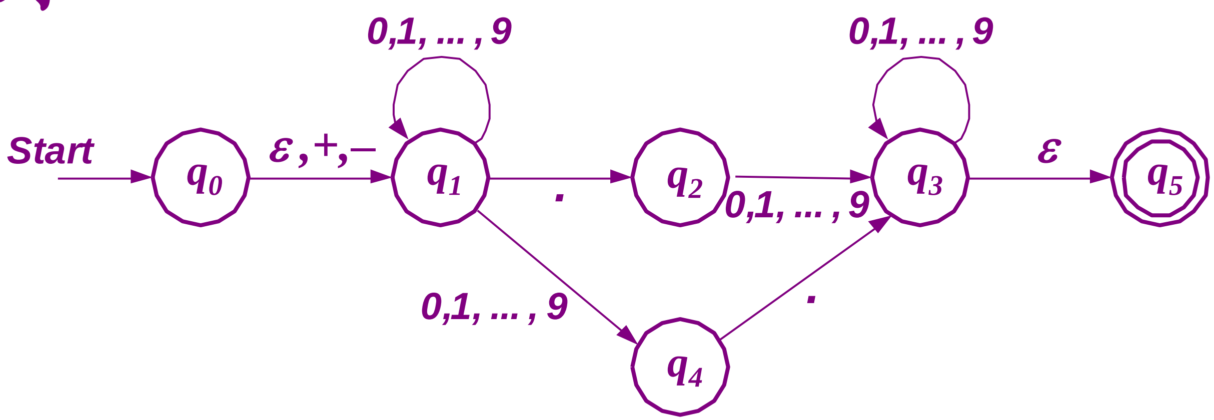
✧ 举例　构造与前面 NFA 等价的 DFA.



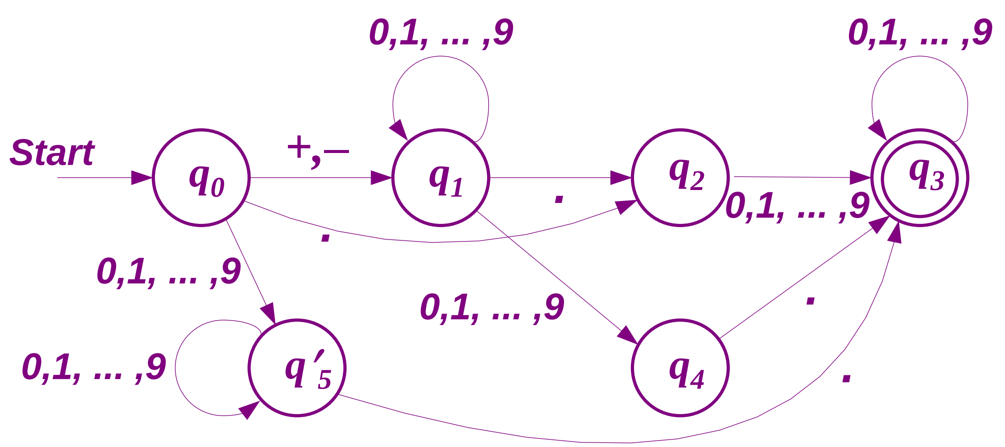


带ε-转移的非确定有限自动机

◇ 举例



比较： *NFA without ε*



带ε-转移的非确定有限自动机

✧ 带ε-转移的非确定有限自动机（ε-NFA）的形式定义

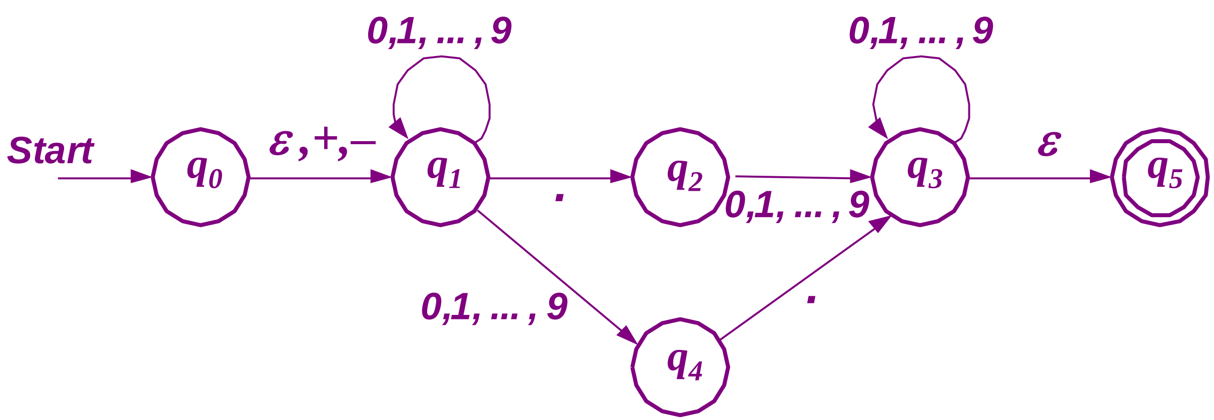
一个 ε-NFA 是一个五元组 $A = (Q, \Sigma, \delta, q_0, F)$.

- 有限状态集 → Q
- 有限输入符号集 → Σ
- 转移函数 → δ
- 一个开始状态 → q_0　　$q_0 \in Q$
- 一个终态集合 → F　　$F \subseteq Q$
- 与 NFA 的不同之处

$$\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$$

带ε-转移的非确定有限自动机

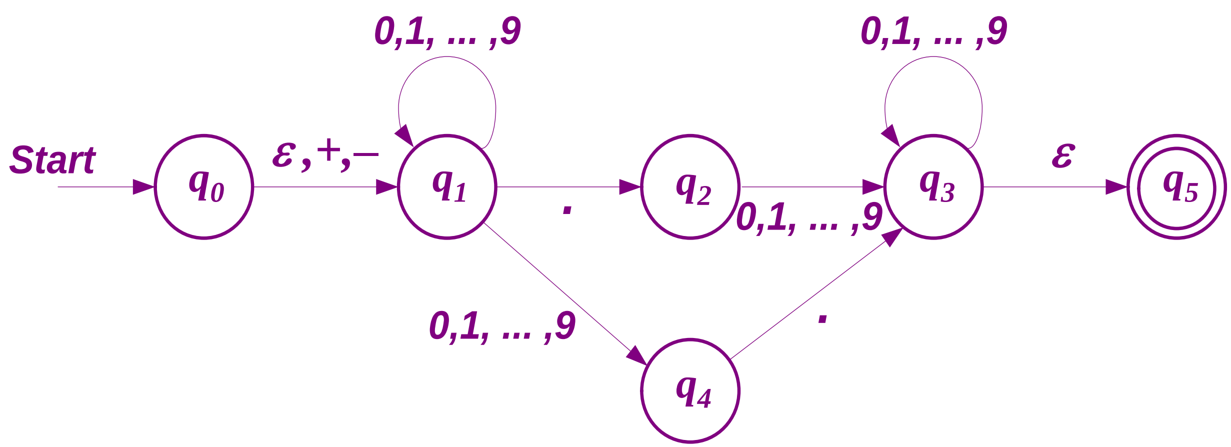
✧ 转移图和转移表表示的ε-NFA



	ε	+,–	.	0,1, … ,9
→ q_0	$\{q_1\}$	$\{q_1\}$	ϕ	ϕ
q_1	ϕ	ϕ	$\{q_2\}$	$\{q_1,q_4\}$
q_2	ϕ	ϕ	ϕ	$\{q_3\}$
q_3	$\{q_5\}$	ϕ	ϕ	$\{q_3\}$
q_4	ϕ	ϕ	$\{q_3\}$	ϕ
* q_5	ϕ	ϕ	ϕ	ϕ

带ε-转移的非确定有限自动机

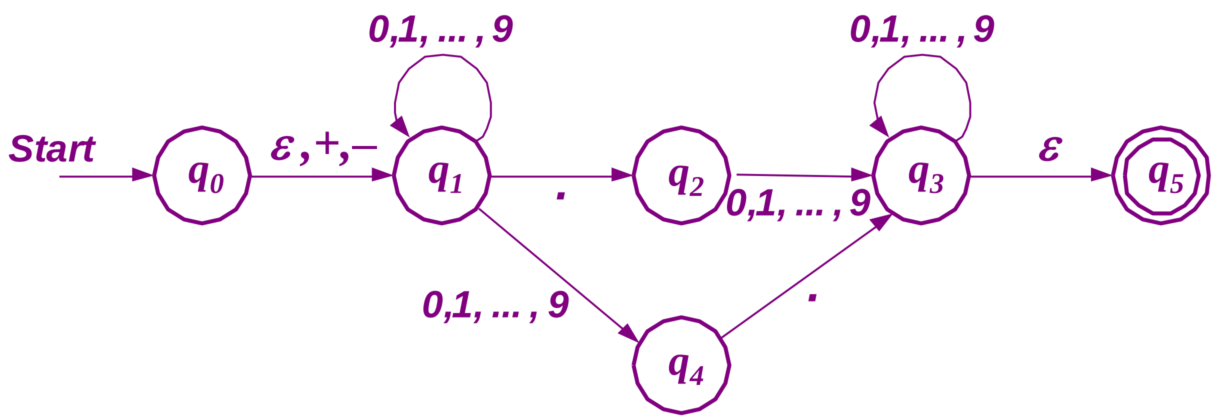
✧ ε - NFA 如何接受输入符号串



– 该 ε - **NFA** 可以接受的字符串如:

- ***3.14***
- +***.314***
- – ***314.***

✧ ε-闭包（closure）



– 状态 q 的ε-闭包，记为**ECLOSE(q)**，定义为从 q 经所有的ε路径可以到达的状态（包括q自身），如:

- *ECLOSE*(q_0) = {q_0,q_1}
- *ECLOSE*(q_2) = {q_2}
- *ECLOSE*(q_3) = {q_3,q_5}

带ε-转移的非确定有限自动机

✧ ε-闭包

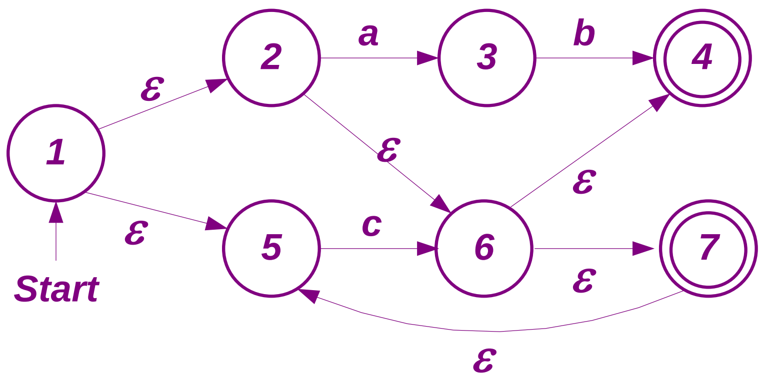
– 设 ε- NFA $A = (Q, \Sigma, \delta, q_0, F)$，$q \in Q$，$ECLOSE(q)$ 为满足如下条件的最小集:

（1）$q \in ECLOSE(q)$

（2）if $p \in ECLOSE(q)$ and $r \in \delta(p, \varepsilon)$, then $r \in ECLOSE(q)$

– 对于右图，有:

- $ECLOSE(1) = \{1,2,4,5,6,7\}$
- $ECLOSE(2) = \{2,4,5,6,7\}$
- $ECLOSE(7) = \{5,7\}$



带ε-转移的非确定有限自动机

✧ 扩展转移函数适合于输入字符串

– 设一个 ε - NFA $E = (Q, \Sigma, \delta, q_0, F)$

– $\delta: Q \times \Sigma \cup \{\varepsilon\} \to 2^Q$

– 扩充定义 δ': $Q \times \Sigma^* \to 2^Q$

– 对任何$q \in Q$，定义:

1 $\delta'(q, \varepsilon) = ECLOSE(q)$

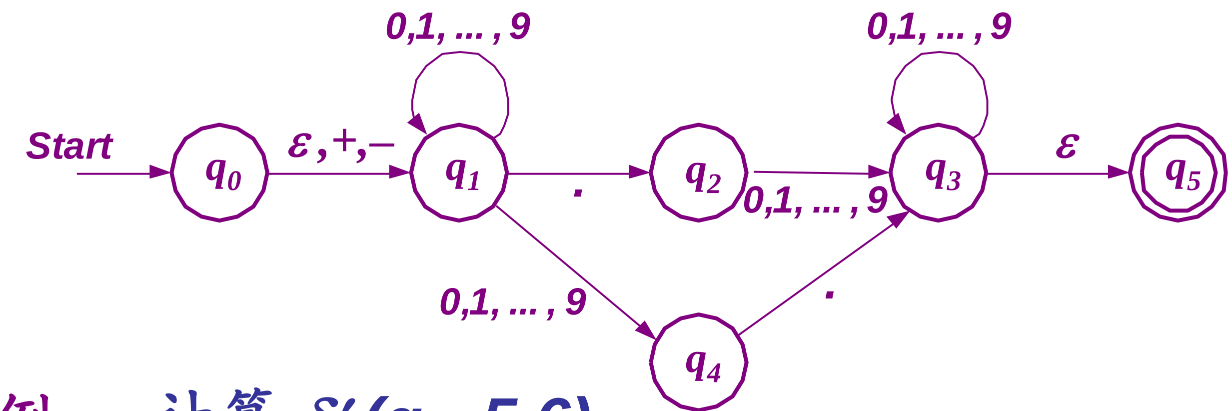
2 若$w = xa$, 其中 $x \in \Sigma^*$, $a \in \Sigma$, 假设

$\delta'(q, x) = \{p_1, p_2, \ldots, p_k\}$, 并且

令 $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \ldots, r_m\}$, 则

$$\delta'(q, w) = \bigcup_{j=1}^{m} ECLOSE(r_j)$$

✧ 扩展转移函数适合于输入字符串



– 举例　计算 $\delta'(q_0, 5.6)$

- $\delta'(q_0, \varepsilon) = ECLOSE(q_0) = \{q_0,q_1\}$
- $\delta(q_0, 5) \cup \delta(q_1, 5) = \{q_1,q_4\}$
  $\delta'(q_0, 5) = ECLOSE(q_1) \cup ECLOSE(q_4) = \{q_1,q_4\}$
- $\delta(q_1, .) \cup \delta(q_4, .) = \{q_2,q_3\}$
  $\delta'(q_0, 5.) = ECLOSE(q_2) \cup ECLOSE(q_3) = \{q_2,q_3,q_5\}$
- $\delta(q_2, 6) \cup \delta(q_3, 6) \cup \delta(q_5, 6) = \{q_3\}$
  $\delta'(q_0, 5.6) = ECLOSE(q_3) = \{q_3,q_5\}$

带ε-转移的非确定有限自动机

✧ ε-NFA 的语言

– 设一个 ε-NFA $E = (Q, \Sigma, \delta, q_0, F)$

– 定义 E 的语言:

$$L(E) = \{ w \mid w \in \Sigma^* \wedge \delta'(q_0, w) \cap F \neq \phi \}$$

– 设 L 是 Σ 上的语言，如果存在一个 ε-NFA $E = (Q, \Sigma, \delta, q_0, F)$，满足 $L = L(E)$，则可以证明 L 也是一个正规语言.

带 ε-转移的非确定有限自动机

✧ ε-NFA 与 DFA 的等价性

定理: L 是某个 ε-NFA 的语言, 当且仅当 L 也是某个 DFA的语言.

证明: 分两步证明.

(1) 设 L 是某个 DFA D 的语言, 则存在一个 ε-NFA E, 满足 $L(E) = L(D) = L$;

(2) 设 L 是某个 ε-NFA E 的语言, 则存在一个 DFA D, 满足 $L(D) = L(E) = L$;

✧ 从 DFA 构造等价的ε- NFA

– 设 L 是某个 DFA $D = (Q, \Sigma, \delta_D, q_0, F)$ 的语言，则存在一个 ε- NFA E，满足 $L(E) = L(D) = L$.

– **证明:** 定义 $E = (Q, \Sigma, \delta_E, q_0, F_E)$，其中 δ_E定义为

- 对任何$q \in Q$, $\delta_E(q, \varepsilon) = \phi$
- 对任何$q \in Q$ 和$a \in \Sigma$,
  若 $\delta_D(q,a) = p$, 则 $\delta_E(q,a) = \{p\}$.

需要证明: 对任何$w \in \Sigma^*$,

$$\delta'_D(q_0,w) = p \text{ iff } \delta'_E(q_0,w) = \{p\}.$$

归纳于 $|w|$ 易证上述命题.

带 ε-转移的非确定有限自动机

✧ 从 ε - NFA 构造等价的 DFA（修改的子集构造法）

– 设 L 是某个 ε - NFA $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ 的语言，则存在一个 DFA D，满足 $L(D) = L(E) = L$.

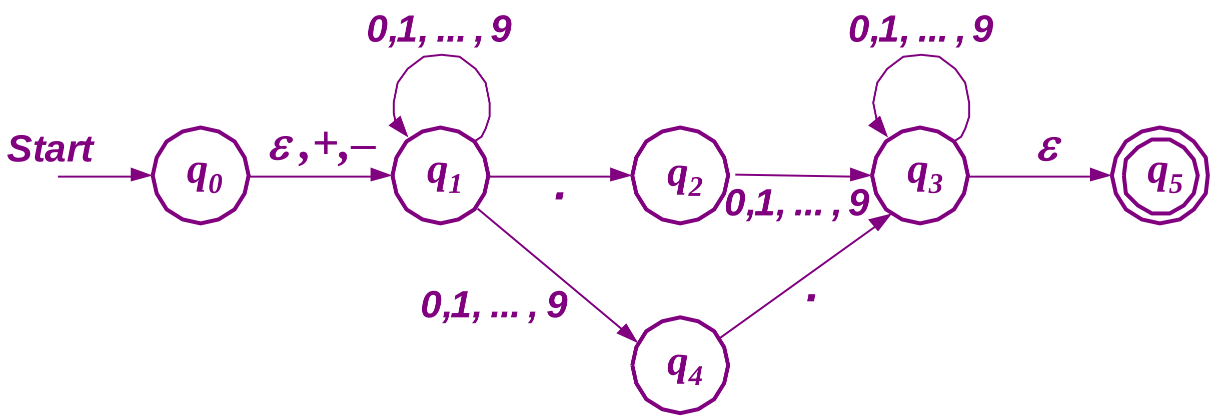
– **证明:** 定义 $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$，其中

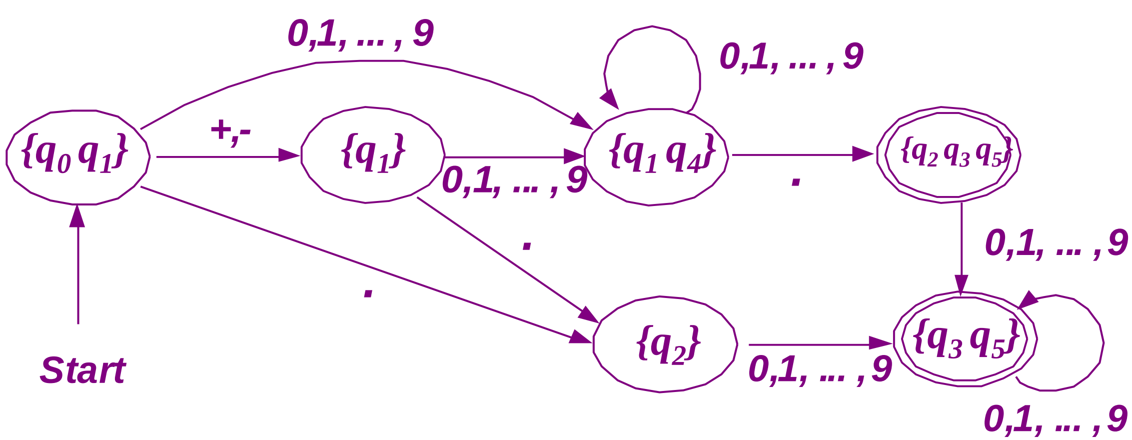
- $Q_D = \{ S \mid S \subseteq Q_E \wedge S = ECLOSE(S) \}$
- $q_D = ECLOSE(q_0)$
- $F_D = \{S \mid S \in Q_D \wedge S \cap F_E \neq \phi \}$
- 对 $S \in Q_D$ 和 $a \in \Sigma$，令 $S = \{ p_1, p_2, \ldots, p_k \}$，并设 $\bigcup_{i=1}^{k} \delta_E(p_i, a) = \{ r_1, r_2, \ldots, r_m \}$，则

$$\delta_D(S, a) = \bigcup_{j=1}^{m} ECLOSE(r_j).$$

需要证明: 对任何 $w \in \Sigma^*$，$\delta'_D(q_D, w) = \delta'_E(q_0, w)$.
归纳于 $|w|$ 可证上述命题.

✧ 修改的子集构造法举例


0,1, … , 9
0,1, … , 9
Start
q0
ε ,+,–
q1
.
q2
0,1, … , 9
q3
ε
q5
0,1, … , 9
.
q4


0,1, … , 9
0,1, … , 9
+,-
{q0 q1}
{q1}
0,1, … , 9
{q1 q4}
.
{q2 q3 q5}
0,1, … ,9
.
.
{q2}
0,1, … ,9
{q3 q5}
Start
0,1, … ,9

带 ε-转移的非确定有限自动机

✧ 从 ε - NFA 构造等价的 DFA（修改的子集构造法）

– 设 $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ 是一个 ε - NFA，通过修改的子集构造法得到相应的DFA $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$, 则
对任何$w \in \Sigma^*$， $\delta'_D(q_D, w) = \delta'_E(q_0, w)$.

– 证明:归纳于 $|w|$

1 设$|w| = 0$, 即 $w = \varepsilon$.
由定义知 $\delta'_D(q_D, \varepsilon) = q_D = ECLOSE(q_0) = \delta'_E(q_0, \varepsilon)$.

2 设$|w| = n+1$, 并 $w = xa$, $a \in \Sigma$. 注意到$|x| = n$.

假设 $\delta'_D(q_D, x) = \delta'_E(q_0, x) = \{p_1, p_2, \dots, p_k\}$.

并设 $\bigcup_{i=1}^{k} \delta_E(p_i, a) = \{r_1, r_2, \dots, r_m\}$.

则 $\delta'_D(q_D, w) = \delta_D(\{p_1, p_2, \dots, p_k\}, a)$

$$= \bigcup_{j=1}^{m} ECLOSE(r_j) = \delta'_E(q_0, w)$$

（确定）有限自动机的最小化

- ✧ 知识回顾：集合上的等价关系与集合的划分
- ✧ *DFA* 状态集合上的一个等价关系
- ✧ 计算状态集划分的算法— 填表法
- ✧ 最小化的 *DFA*

✧ 知识回顾：集合上的等价关系与集合的划分

– 等价关系

设 Q 为一个集合，二元关系 R 是 Q 上的一个等价关系，当且仅当满足以下条件：

1. **自反性** 对任何$a \in Q$, aRa 成立；
2. **对称性** 对任何$a,b \in Q$, 如果 aRb 成立，则有 bRa 成立；
3. **传递性** 对任何$a,b,c \in Q$, 如果 aRb 和 bRc 成立，则有 aRc 成立.

✧ 知识回顾：集合上的等价关系与集合的划分

– 等价关系与划分

设 Q 为一个集合，R 是 Q 上的一个等价关系，由 R 产生的所有等价类（或块）的集合构成 Q 的一个划分.

– 解释

1. 等价类 对任何 $a \in Q$, a 所在的块用 $[a]$ 表示, 定义为

$$[a] = \{x \mid xRa\};$$

2. 每一元素都属于唯一的块 即满足

（1）$\cup_{a \in Q} [a] = Q$； 和

（2）对任何 $a,b \in Q$, 或者 $[a]=[b]$, 或者 $[a]\cap[b]=\Phi$

✧ DFA 状态集合上的一个等价关系

– 设一个 DFA $D = (Q, \Sigma, \delta, q_0, F)$, 定义Q上的一个二元关系 R 为： 对任何 $p,q \in Q$,

$$pRq \text{ iff } \forall w \in \Sigma^*. (\hat{\delta}(p,w) \in F \leftrightarrow \hat{\delta}(q,w) \in F)$$

– 结论 上述关系 R 是等价关系.

证明： 1. 自反性 对任何$q \in Q$, qRq 成立;

2. 对称性 对任何$p,q \in Q$, $pRq \rightarrow qRp$ 成立;

3. 传递性 对任何$p,q,r \in Q$, 设pRq 和 qRr 成立, 即对任何$w \in \Sigma^*$, $\hat{\delta}(p,w) \in F \leftrightarrow \hat{\delta}(q,w) \in F$ 和 $\hat{\delta}(q,w) \in F \leftrightarrow \hat{\delta}(r,w) \in F$ 成立; 由此, 也有 $\hat{\delta}(p,w) \in F \leftrightarrow \hat{\delta}(r,w) \in F$ 成立. 所以, qRr 成立

□

（确定）有限自动机的最小化

✧ DFA 状态集合上的一个等价关系

– 若pRq, 称 p 和 q 等价（equivalent）. 若p 和q 不等价，则称 p 和q 是可区别的（distinguashable）.

– 关系 R 对应有限状态集 Q 的一个划分;
该划分的每个块是 Q 的一个子集;
同一划分块中的所有状态之间都是相互等价的;
分属不同划分块的任何两个状态之间都是可区别的.

✧ DFA 的优化

通过合并等价的（或不可区别的）状态
关键：如何计算上述划分?

✧ 有关可区别性的几个有用的结果

– 设状态 *r* 和 *s* 通过某个输入符号 *a* 可分别转移到 *p* 和 *q*（即 $\delta(r,a)=p$, $\delta(s,a)=q$），则有

p 和 q 可区别 $\Rightarrow$ r 和 s 可区别

这是因为：若 *p* 和 *q* 可为字符串 *w* 区别，则 *r* 和 *s* 可为字符串 *aw* 区别.

（ $\because$ $\delta'(r,aw) = \delta'(p,w)$, $\delta'(s,aw) = \delta'(q,w)$ ）

✧ 有关可区别性的几个有用的结果

– 设状态 r 和 s 通过某个输入符号 a 可分别转移到 p 和 q（即 $\delta(r,a)=p$, $\delta(s,a)=q$），则有

r 和 s 不可区别 $\Rightarrow$ p 和 q 不可区别

（ 前页结果的逆否 ）

✧ 有关可区别性的几个有用的结果

– 设状态 r 和 s 通过某个输入符号 a 可分别转移到 p 和 q（即 $\delta(r,a)=p$, $\delta(s,a)=q$），则有

r 和 s 可由 ax 区别 $\Rightarrow$ p 和 q 可由 x 区别

（$\because$ $\delta'(r,ax)=\delta'(p,x)$, $\delta'(s,ax)=\delta'(q,x)$）

✧ 计算状态集划分的算法— 填表法

– 填表算法（*table-filling algorithm*）基于如下递归地标记可区别的状态偶对的过程:

基础 如果 *p* 为终态，而 *q* 为非终态，则 *p* 和 *q* 标记为可区别的;

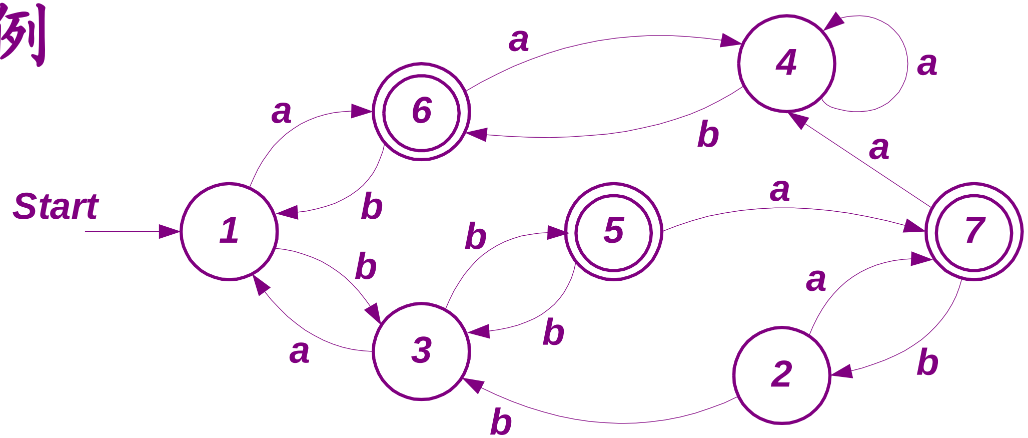
归纳 设 *p* 和 *q* 已标记为可区别的, 如果状态 *r* 和 *s* 通过某个输入符号 *a* 可分别转移到 *p* 和 *q*，$\delta(r,a)=p$, $\delta(s,a)=q$, 则 *r* 和 *s* 也标记为可区别的;

（确定）有限自动机的最小化

✧ 计算状态集划分的算法—填表法

– 填表算法举例

2						
3	x	x				
4	x	x	x			
5	x	x	x	x		
6	x	x	x	x	x	
7	x	x	x	x	x	
	1	2	3	4	5	6



(1) 区别所有终态和非终态

(2) 区别(1,3), (1,4), (2,3), (2,4), (5,6), (5,7)

(3) 区别 (3,4)

(4) 结束. 划分结果：{1,2}, {3}, {4}, {5}, {6,7}

✧ 计算状态集划分的算法—填表法

– **填表算法的正确性** 还需证明：如果两个状态没有被填表算法标记，则这两个状态一定是等价的

– **证明** 反证法. 假定状态 r 和 s 没有被填表算法标记，但这两个状态不是等价的，即是可区别的.

设字符串 w 可用于区别状态 r 和 s，即 $\delta(r,w)$ 和 $\delta(s,w)$ 两个状态中，一个是终态，一个是非终态. 不妨设前者为终态, 后者为非终态.

首先不可能有 $w=\varepsilon$, 否则, 状态 r 为终态, 而 s 为非终态，依填表算法, r 和 s 第一步就被标记.

设 $w=ax$, 并且 $\delta(r,a)=p$, $\delta(s,a)=q$, 则 p 和 q 可被 x 区别. 但同样 p 和 q 不可能被填表算法标记(否则, r 和 s 将被标记). 同样也有, $x\neq\varepsilon$.

该过程不可能一直下去, 终将产生矛盾.

✧ 通过合并等价的状态进行 *DFA* 的优化

– 步骤

1. 删除所有从开始状态不可到达的状态及与其相关的边, 设所得到的 *DFA* 为 $A = (Q, \Sigma, \delta, q_0, F)$;

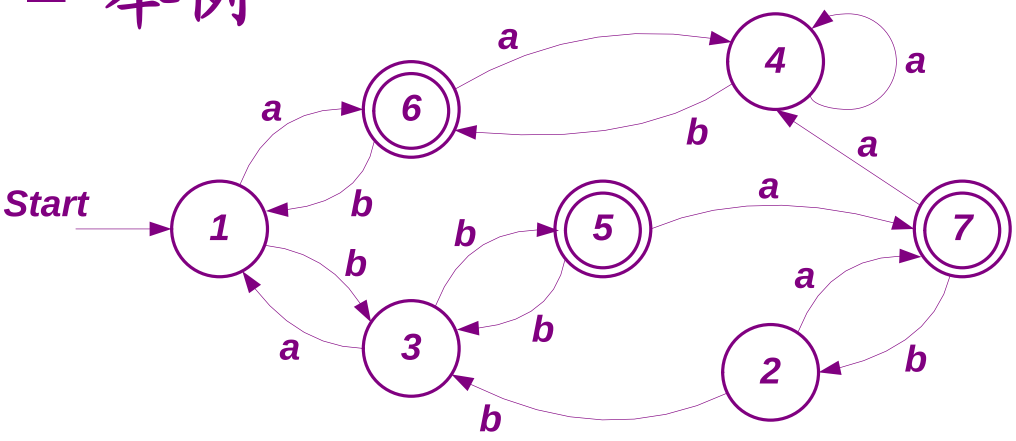
2. 使用填表算法找出所有等价的状态偶对;

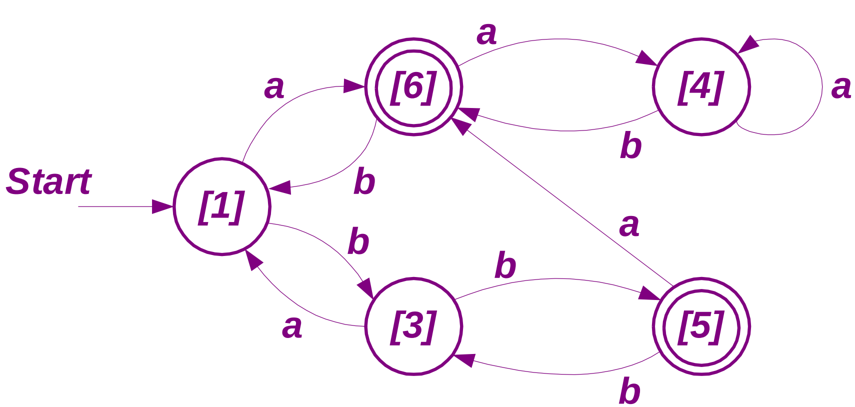
3. 根据 2 的结果计算当前状态集合的划分块，每一划分块中的状态相互之间等价，而不同划分块中的状态之间都是可区别的. 包含状态 q 的划分块用 $[q]$ 表示.

4. 构造与 *A* 等价的 *DFA* $B = (Q_B, \Sigma, \delta_B, [q_0], F_B)$, 其中 $Q_B=\{ [q] \mid q \in Q\}$, $F_B = \{ [q] \mid q \in F\}$, $\delta_B([q], a)=[\delta(q,a)]$

✧ 通过合并等价的状态进行 *DFA* 的优化

– 举例





– 等价的状态偶对为：

（*1, 2*），（*6, 7*）

– 划分结果：

{ 1, 2 }, {3}, {4}, {5}, { 6, 7 }

– 新的状态集合：

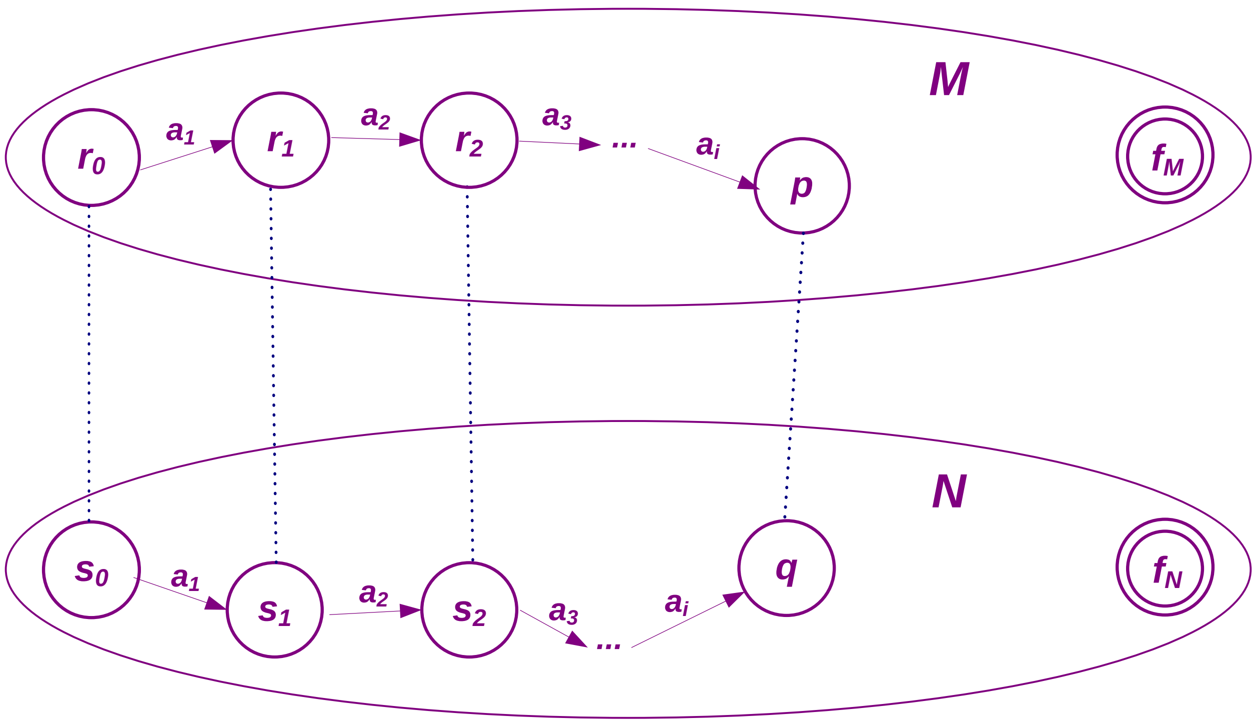
[1], [3], [4], [5], [6]

✧ 最小化的 DFA

– 问题 假定一个 DFA 为 A, 用上述优化步骤构造出与 A 等价的DFA M; 那么是否存在一个状态数目比 M 还少的DFA N, 它接受的语言同 A 和 M 完全一样?

假设存在一个这样 DFA N. 现将 M 和 N 相并, 即状态、转移规则都相并, 这里假定 M 和 N 之间没有重名的状态, 因而也没有相交的转移边, 原来的终态还是终态, 原来的两个初态中任选一个作为新的初态. 同时还假定 M 和 N 的每一状态都是从其相应的初态可以到达的, 否则我们将去掉不可达状态, 得到状态数目更小的 DFA.

✧ 最小化的 *DFA*

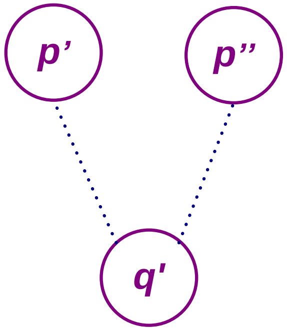


✧ 最小化的 DFA

对 *M* 和 *N* 相并后的 *DFA* 运用填表算法可以得出:

1. *M* 和 *N* 的初态是不可区分的, 因为 *L(M)=L(N)*;
2. 若 *r* 和 *s* 是不可区分的, 则对于任何输入符号, *r* 和 *s* 的后继状态之间也是不可区分的;
3. *M* 的任一状态至少与 *N* 的一个状态是不可区分的.

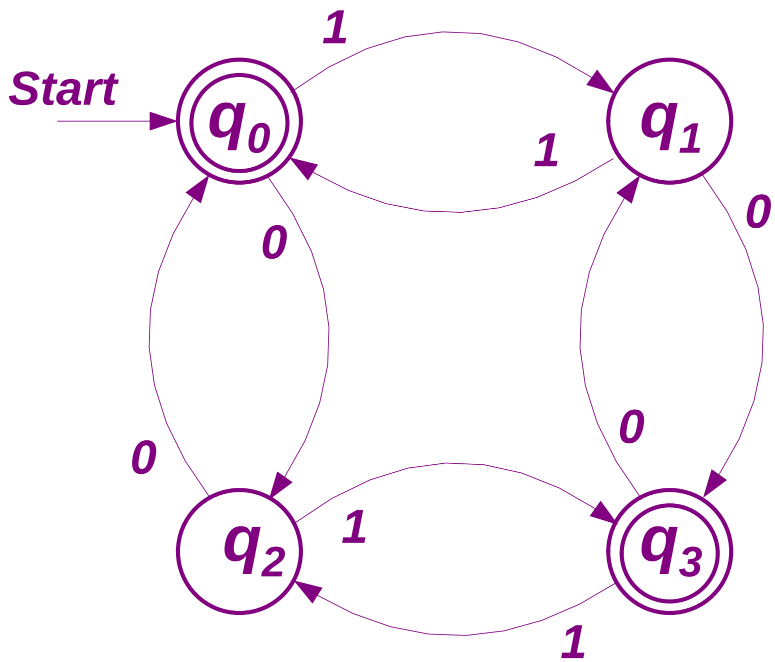
根据假设, *N* 的状态数目比 *M* 少, 所以 *M* 中必然存在两个状态, 它们分别与 *N* 中的同一个状态不可区分. 根据不可区分关系的传递性, *M* 的这两个状态是不可区分的, 这与 *M* 的构造过程矛盾.



– 结论 对任何 *DFA A*, 用前述优化步骤构造出与 *A* 等价的 *DFA M*; 那么 *M* 的状态数目不多于任何语言为 *L(A)* 的 *DFA*

✧ 课堂练习

– 最小化下图表示的 DFA



课后练习

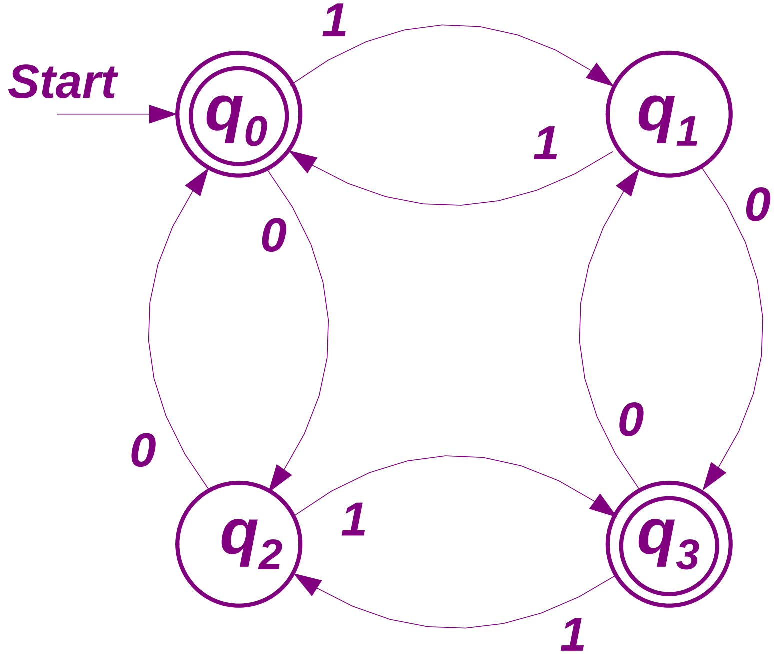
- 必做题:
  - *Ex.2.2.2
  - Ex.2.2.4 (b),(c)
  - Ex.2.2.5 (d)
  - Ex.2.2.7
  - Ex.2.3.2
  - Ex.2.3.4 (b),(c)
  - Ex.2.4.2 (c) (请依所介绍的算法做)
  - Ex.2.5.2
  - Ex.2.5.3 (a), ! (b)
  - Ex.4.4.2
- 思考题:
  - Ex.2.2.5 (b)
  - Ex.2.2.6 *(a),(b)
  - Ex.2.5.3 (c)

课后练习

✧ 思考题:

– 附加

Let L={ w | In w, the number of 0's and the number of 1's are the same in parity } be a language over Σ={0,1} . Prove that L is the language of following DFA.



课后练习

- **自测题:**
  - 设计一个 DFA，其语言为长度至少为2且头两个字符不相同的0，1串构成的集合。
  - 试构造接受下列语言的一个 DFA:

    1）$\{w \in \{a, b\}^* \mid w$ 中不包含子串 $aa\}$
    2）$\{w \mid w \in \{a, b\}^*$，$w$ 中包含且仅包含奇数个子串 $ab\}$
    3）$\{w \mid w \in \{a, b\}^*$，且 w 中 a 的个数和 b 的个数之和是奇数$\}$
    4）$\{w \mid w \in \{a,b\}^*$，w 中 a 的个数是偶数，且 w 的长度也为偶数$\}$
    5）$\{w \mid w \in \{a, b\}^*$，w 包含子串 ab，但不包含子串 bb$\}$
  - 设计一个NFA，其语言为长度至少为2且末尾两个字符不相同的0，1串构成的集合。
  - 试给出下列正规语言 L 的一个 NFA（不是 ε-NFA）：

    $\{w \mid w \in \{a, b\}^*,\ |w| \geq 3$，且 w 中第2位和第3位不同$\}$

课后练习

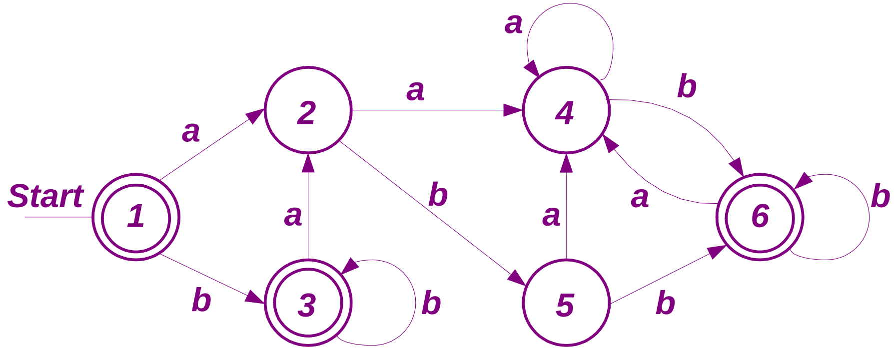
- 自测题:
  - 设计一个ε–NFA，其语言由满足下述条件的0，1串构成：长度至少为1，且前三个符号中至少有一个“0”。（以状态转移图的形式给出，要能够体现ε–NFA的设计特点）。
  - 试构造接受下列正规语言 *L* 的一个ε-*NFA*:

    1) $\{ w \mid w \in \{a, b, c\}^*, |w| \geq 1,$ 且 w 后 3 位中至少有一位不是 $c \}$

    2) $\{ w \mid w \in \{a, b, c\}^*, |w| \geq 2,$ 且 w 中从第 2 位到第4位至少有一位不是 c $\}$

课后练习

- ✧ 自测题:
  - – 左下图表示一个 DFA，构造出与该DFA等价的最小化的DFA（即拥有的状态数目最少）。(分主要步骤或直接写出结果均可)



That's all for today.

Thank You