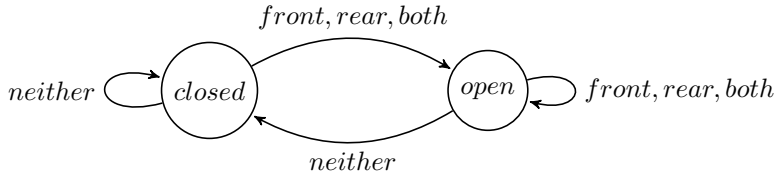
# 2 Automata and Language

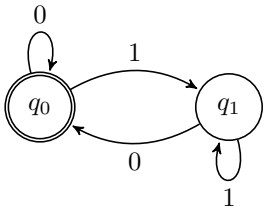
## 2.1 Finite Automaton

***e.g.* 2.1.** *Automatic Door*




| | *front* | *rear* | *both* | *neither* |
|---|---|---|---|---|
| *front* | ✓ | × | ✓ | × |
| *rear* | × | ✓ | ✓ | × |

***e.g.* 2.2.** $L = \{w \in \{0,1\}^* | w = w_1 w_2 \cdots w_n, w_n = 0\}$




***remark.*** $q_0$ : *accepted state*

$Q = \{q_0, q_1\}, \Sigma = \{0,1\}, \delta : Q \times \Sigma \to Q$

| | 0 | 1 |
|---|---|---|
| $q_0$ | $q_0$ | $q_1$ |
| $q_1$ | $q_0$ | $q_1$ |

***def* 2.1.** *(finite automaton)*

*A* ***finite automaton*** *is a 5-tuple* $(Q, \Sigma, \delta, q_0, F)$*, where:*

1. $Q$ *is a finite set called the states*
2. $\Sigma$ *is the alphabet*

3. $\delta : Q \times \Sigma \to Q$ *is the transition function*

4. $q_0$ *is the start state*

5. $F \subseteq Q$ *is the set of accept states*

***def* 2.2.** *Let $M = (Q, \Sigma, \delta, q_0, F)$ be a finite automaton, let $w = w_1 w_2 \cdots w_n$ be a string, where each $w_i \in \Sigma$.*

*Then M accept w if there is a sequence of states $r_0, r_1, \cdots, r_n \in Q$, such that:*

1. $r_0 = q_0$

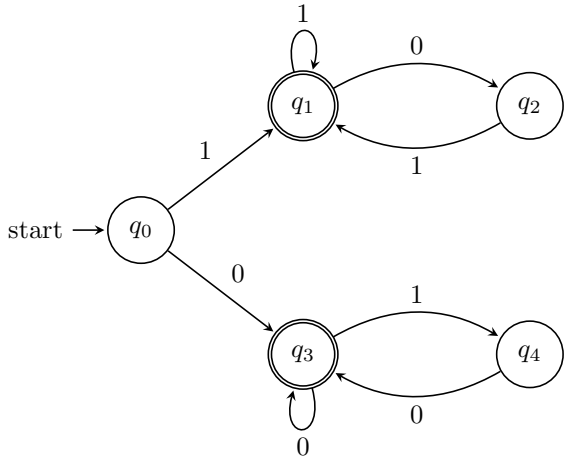
2. $\delta(r_i, w_{i+1}) = r_{i+1}$, *for* $i = 0, 1, 2, \cdots, n-1$

3. $r_n \in F$

***def* 2.3.** *If L is the set of strings that M accepts, we say L is the language of M, and write L(M) = L, we say M recognizes/decides/accepts L.*

*If M accepts no string, it recognizes one language namely, the empty language.*

***e.g.* 2.3.** $L = \{w \in \{0,1\}^* | w = w_1 w_2 \cdots w_n, w_n = w_1\}$




## 2.2 Regular Language

***def* 2.4.** *(regular language) $L \subseteq \Sigma^*$ is a **regular language** if there is a finite automaton that accepts L*

*Let $A, B \subseteq \Sigma^*$, define:*

- *(union)* $A \cup B = \{x \in \Sigma^* | x \in A \text{ or } x \in B\}$

- *(concatenation)* $AB = \{xy | x \in A, y \in B\}$
- *(star)* $A^* = \{x_1 x_2 \cdots x_k | k \geq 0, x_1, x_2, \cdots, x_k \in A\}$

***thm* 2.5.** *If $A_1, A_2$ are regular languages, so is $A_1 \cup A_2$*

*Proof. Let $M_1 = (Q_1, \Sigma_1, \delta_1, q_{10}, F_1)$ accepts $A_1$, $M_2 = (Q_2, \Sigma_2, \delta_2, q_{20}, F_2)$ accepts $A_2$, construct $M = (Q, \Sigma, \delta, q_0, F)$:*

1. $Q = Q_1 \times Q_2 = \{(r_1, r_2) | r_1 \in Q_1, r_2 \in Q_2\}$
2. $\delta : Q \times \Sigma \to Q$ *is defined as for each* $(r_1, r_2) \in Q$, *and each* $a \in \Sigma$, *let* $\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$
3. $q_0 = (q_{10}, q_{20})$
4. $F = \{(r_1, r_2) | r_1 \in F_1 \text{ or } r_2 \in F_2\}$

□

***remark.*** *so is $A \cap B = \overline{\overline{A} \cup \overline{B}}$*[1]

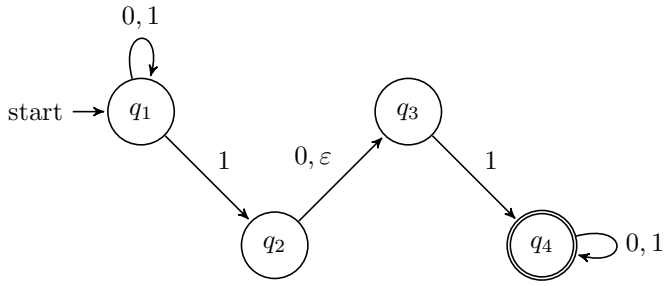
## 2.3 DFA and NFA

***thm* 2.6.** *If $A_1, A_2$ are regular languages, so is $A_1 A_2$*

- ***DFA****: deterministic finite automaton*
- ***NFA****: nondeterministic...*

***If at least one of these processes accepts, then the entire computation accepts.***

***e.g.* 2.4.** *NFA:*




*input:* 010110

[1] *proof of the closure under complement will be mentioned later*

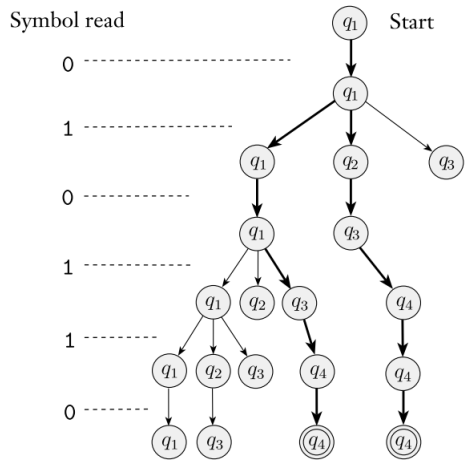



Figure 1: *The computation of NFA on input 010110*

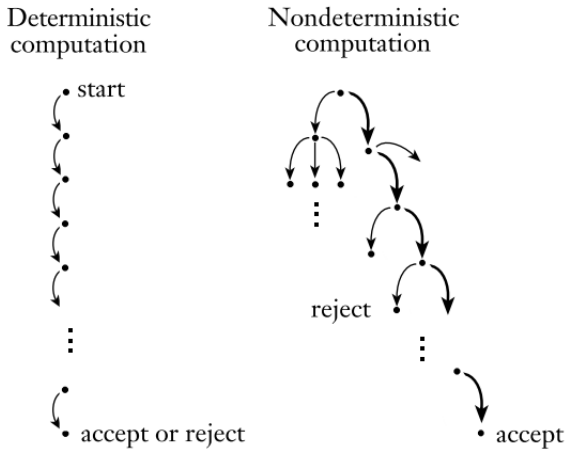
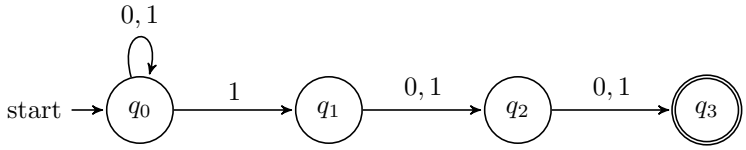


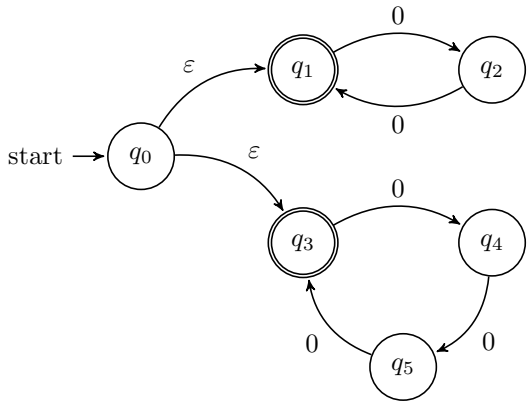

Figure 2: *Deterministic and nondeterministic computations with an accepting branch*

***remark.*** *If a language can be accepted by DFA, then its time complexity is O(n), space complexity is O(1).*

***e.g.* 2.5.** *Let A be the language consisting of all strings over* $\{0,1\}$ *containing a* 1 *in the third position from the end (e.g.,* 000100 *is in A but* 0011 *is not). The following four-state NFA recognizes A.*




***e.g.* 2.6.** $L = \{0^k, 2 | k \text{ or } 3 | k\}$




***def* 2.7.** *(NFA)*

*An* ***NFA*** *is a 5-tuple* $(Q, \Sigma, \delta, q_0, F)$*,where:*

1. $Q$ *is a finite set of states*
2. $\Sigma$ *is the alphabet*
3. $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to P(Q)$ *is the transive function*
4. $q_0 \in Q$ *is the start state*
5. $F \subseteq Q$ *is the set of accept states*

***remark.*** $P(Q)$ *is the collection of all the subsets of Q (power set)*

- *variant1:* $\delta : Q \times \Sigma^* \to Q$

  *We guarantee there is at most one applicable transition.*

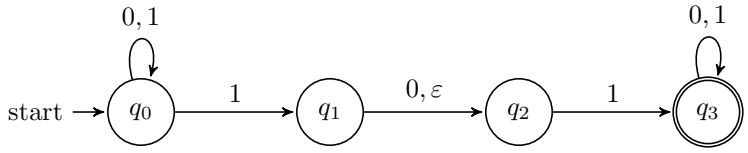
- *variant2: If there are multiple applicable transitions,non-deterministically choose one.*

***def* 2.8.**

*Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA, and let $w \in \Sigma^*$. Say $N$ accepts $w$ if we can write $w = y_1 y_2 \cdots y_m$, where $y_i \in \Sigma \cup \{\varepsilon\}$, and there exist $r_0, r_1, \cdots, r_m \in Q$, such that:*

1. $r_0 = q_0$
2. $r_{i+1} \in \delta(r_i, y_{i+1})$ *for* $i = 0, 1, \cdots, m-1$
3. $r_m \in F$

***e.g.* 2.7.** $Q = \{q_0, q_1, q_2, q_3\}, \Sigma = \{0, 1\}, F = \{q_3\}, \delta : Q \times \Sigma \to Q$




| $q \backslash \Sigma$ | 0 | 1 | $\varepsilon$ |
|---|---|---|---|
| $q_0$ | $\{q_0\}$ | $\{q_0, q_1\}$ | $\emptyset$ |
| $q_1$ | $\{q_2\}$ | $\emptyset$ | $\{q_2\}$ |
| $q_2$ | $\emptyset$ | $\{q_3\}$ | $\emptyset$ |
| $q_3$ | $\{q_3\}$ | $\{q_3\}$ | $\emptyset$ |

***thm* 2.9.** *Every NFA has an equivalent DFA*

*Proof. Let $N = (Q, \Sigma, \delta, q_0, F)$ be the NFA recognizing $A$. Construct a DFA $M = (Q', \Sigma, \delta', q_0', F)$ recognizing $A$:*

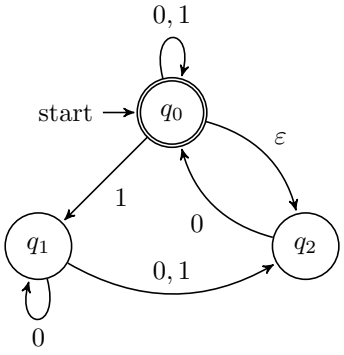
*Let $E(R) = \{q \in Q |$ $q$ can be reached from $R$ by traveling along zero or more $\varepsilon$ arrows$\}$*

*Define $M$ as follows:*

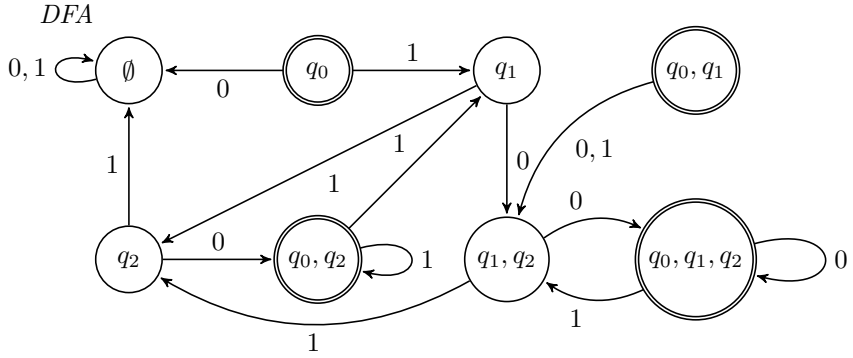
1. $Q' = P(Q)$
2. *For $R \in Q'$ and $a \in \Sigma$, let $\delta'(R, a) = \{q \in Q : q \in E(\delta(r, a))$ for some $r \in R\} = \bigcup_{r \in R} E(\delta(r, a))$*
3. $q_0' = E(\{q_0\})$
4. $F' = \{R \in Q', R \cap F \neq \emptyset\}$

□

***e.g.* 2.8.** *NFA*




*DFA*




***corollary* 2.10.** *A language is a regular language iff an NFA recognizes it.*

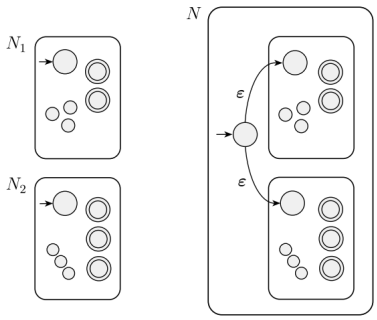



Figure 3: *Construction of an NFA $N$ to recognize $A_1 \cup A_2$*

***thm* 2.5.** *The class of regular languages is closed under union.*

*Second proof*

*Proof. See figure 3.*

*Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recignize $A_1$ and $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize $A_2$*

*Construct $N = (Q, \Sigma, \delta, q, F)$ as follows:*

1. $Q = Q_1 \cup Q_2 \cup \{q_0\}$
2. $q_0$ *is the start state*
3. $F = F_1 \cup F_2$
4. *For* $q \in Q, a \in \Sigma \cup \{\varepsilon\}$.

$$\text{Let } \delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q \\ \delta_2(q,a) & q \in Q \\ \{q_1, q_2\} & q = q_0 \text{ and } a = \varepsilon \\ \emptyset & q = q_0 \text{ and } a \neq \varepsilon \end{cases}$$

□

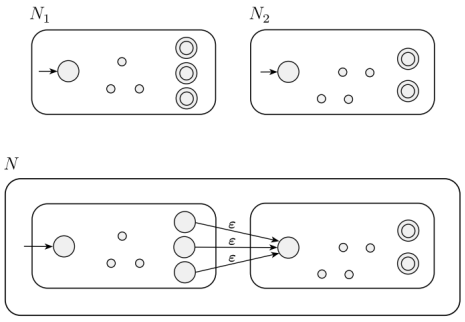



Figure 4: *Construction of an NFA $N$ to recognize $A_1A_2$*

***thm* 2.11.** *The class of regular languages is closed under concatenation.*

*Proof. See figure 4. Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recignize $A_1$ and $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize $A_2$*

*Construct $N = (Q, \Sigma, \delta, q, F)$ as follows:*

1. $Q = Q_1 \cup Q_2$
2. $q_1$ *is the start state*
3. $F = F_2$
4. *For $q \in Q, a \in \Sigma \cup \{\varepsilon\}$.*

$$\text{Let } \delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & q \in F_1 \text{ and } a = \varepsilon \\ \delta_2(q,a) & q \in Q_2 \end{cases}$$

□

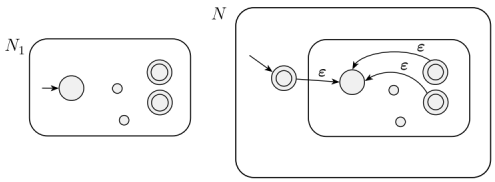



Figure 5: *Construction of an NFA N to recognize* $A_1^*$

***thm* 2.12.** *The class of regular languages is closed under star.*

*Proof. See figure 5. Let* $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ *recognize* $A_1$

*Construct* $N = (Q, \Sigma, \delta, q, F)$ *as follows:*

1. $Q = Q_1 \cup \{q_0\}$
2. $q_0$ *is the start state*
3. $F = \{q_0\} \cup F_1$
4. *For* $q \in Q, a \in \Sigma \cup \{\varepsilon\}$.

$$\text{Let } \delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \text{ and } a = \varepsilon \\ \{q_1\} & q = q_0 \text{ and } a = \varepsilon \\ \emptyset & q = q_0 \text{ and } a \neq \varepsilon \end{cases}$$

□

***thm* 2.13.** *The class of regular languages is closed under complement.*

*Proof. Let DFA* $M = (Q, \Sigma, \delta, q_0, Q_{accept})$ *recognizing A, then* $\overline{A} = \Sigma^* - A$,
*construct* $M' = (Q, \Sigma, \delta, q_0, Q'_{accept}), Q'_{accept} = Q - Q_{accept}$, *it's easy to prove* $L(Q') = \overline{A}$. □

## 2.4 Regular Expression

***e.g.* 2.9.** *Consider students' ID number, with boys' ending with an odd number, while girls' ending with even number.*

*boys* : $(0 \cup 1 \cup \cdots \cup 9)^*(1 \cup 3 \cup 5 \cup 7 \cup 9)$

***def* 2.14.** *(regular expression)*

*$R$ is a* ***regular expression*** *if $R$ is:*

1. *$a$ for some $a \in \Sigma$*
2. *$\varepsilon$*
3. *$\emptyset$*
4. *$R_1 \cup R_2$, where $R_1, R_2$ are regular expressions*
5. *$R_1 R_2$, where $R_1, R_2$ are regular expressions*
6. *$R^*$, where $R$ is a regular expression*

***e.g.* 2.10.**

1. $0^*10^*$
2. $\Sigma^*1\Sigma^* = \{w | w = \cdots 1 \cdots\}$
3. $\Sigma^*001\Sigma^* = \{w | w = \cdots 001 \cdots\}$
4. $1^*(01^+)^* = \{w \in \{0,1\}$, *every 0 in $w$ is followed by at least 1*$\}$
5. $(\Sigma\Sigma)^* = \{w |$ *the length of $w$ is even*$\}$
6. $1^*\emptyset = \emptyset$

***exercises* 2.1.**

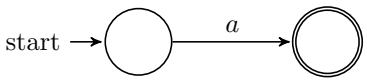
1. *$w$ contains substring 110:* $\{\Sigma^*110\Sigma^*\}$
2. *$w$ doesn't contain 00 as substring:* $(0 \cup \varepsilon)(1 \cup 10)^*$
3. *the number of 1s is a multiple of 3:* $\{(0^*10^*10^*10^*)^*\}$
4. *$w$ contains at least two 1s and one 0:* $(\Sigma^*1\Sigma^*1\Sigma^*0\Sigma^*) \cup (\Sigma^*1\Sigma^*0\Sigma^*1\Sigma^*) \cup (\Sigma^*0\Sigma^*1\Sigma^*1\Sigma^*)$

***thm* 2.15.** *A language is regular iff some regular expression described it*

***lemma* 2.16.** ($\Leftarrow$) *If a language is decribed by a regular expression, then it's regular*

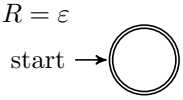
*Proof. Let's convert $R$ into an NFA $N$*

1. $R = a, a \in \Sigma$

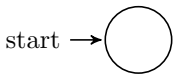
   *Note that this machine fits the definition of an NFA but not that of a DFA because it has some states with no exiting arrow for each possible input symbol. Of course, we could have presented an equivalent DFA here; but an NFA is all we need for now, and it is easier to describe.*
   *Formally, $N = (\{q_1, q_2\}, \Sigma, \delta, q_1, \{q_2\})$, where we describe $\delta$ by saying that $\delta(q_1, a) = \{q_2\}$ and that $\delta(r, b) = \emptyset$ for $r \neq q_1$ or $b \neq a$.*

2. $R = \varepsilon$

   *Formally, $N = (\{q_1\}, \Sigma, \delta, q_1, \{q_1\})$, where we describe $\delta$ by saying that $\delta(q_1, a) = \emptyset$ for $\forall r, b$.*

3. $R = \emptyset$

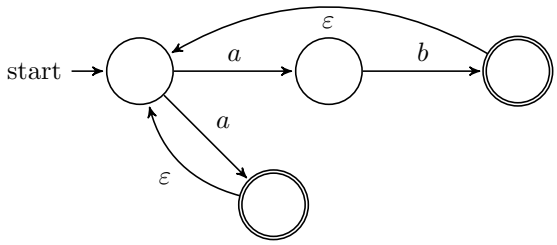
   *Formally, $N = (\{q_1\}, \Sigma, \delta, q_1, \emptyset)$, where we describe $\delta$ by saying that $\delta(q_1, a) = \emptyset$ for $\forall r, b$.*

4. $R = R_1 \cup R_2$
5. $R = R_1 R_2$
6. $R = R_1^*$

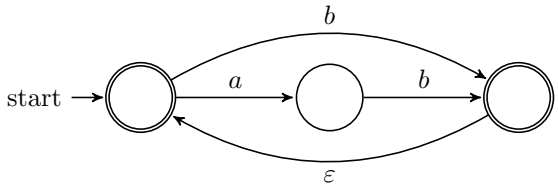
   *For the last three cases, we use the constructions given in the proofs that the class of regular languages is closed under the regular operations. In other words, we construct the NFA for $R$ from the NFAs for $R_1$ and $R_2$ (or just $R_1$ in case 6) and the appropriate closure construction.*

□

***e.g.* 2.11.** $(ab \cup a)^*$




*It can be simplified as below:*




*Or as this:*