

# Chap. 3 Regular Expressions and Languages

## 3.1.2 Regular Expressions over some alphabet $T$ .

**Basis:**

1. The constant  $\varepsilon$  is a regular expression, denoting the languages  $\{\varepsilon\}$ , i.e.,  $L(\varepsilon) = \{\varepsilon\}$ .
2. The constant  $\emptyset$  is a regular expression, denoting the languages  $\emptyset$ , i.e.,  $L(\emptyset) = \emptyset$ .
3. If  $a \in T$ , then  $\mathbf{a}$  is a regular expression, denoting the languages  $\{a\}$ , i.e.,  $L(\mathbf{a}) = \{a\}$ .

**Induction:**

1. If  $E$  and  $F$  are regular expressions, then  $E + F$  is a regular expression, denoting union of  $L(E)$  and  $L(F)$ , i.e.,  $L(E + F) = L(E) \cup L(F)$ .
2. If  $E$  and  $F$  are regular expressions, then  $EF$  is a regular expression, denoting concatenation of  $L(E)$  and  $L(F)$ ,

*i.e.*,  $L(EF) = L(E)L(F)$ .

3. If  $E$  is a **regular expression**, then  $E^*$  is a **regular expression**, denoting **closure** of  $L(E)$ , *i.e.*,  $L(E^*) = (L(E))^*$ .
4. If  $E$  is a **regular expression**, then  $(E)$  is a **regular expression**, denoting  $L(E)$ , *i.e.*,  $L((E)) = L(E)$ .

*Example 3.2 in p.89*

### **3.1.3 Precedence of Regular Expression Operators**

0. *parenthesis*

1. *closure(\*)*

2. *concatenation(·) or juxtaposed.*

3. *union(+)*

*Example 3.3 in p.91*

### **Equivalence of regular expressions**

*Let  $R, S$  be regular expressions. We say  $R = S$ , if  $L(R) = L(S)$ .*

## 3.2 Finite Automata and Regular Expressions

### 3.2.1 From DFA's to Regular Expressions

**Theorem 3.4** *If  $L = L(D)$  for some DFA  $D$ , then there is a regular expressions  $R$  such that  $L = L(R)$ .*

**Proof** *Let us suppose  $D$ 's states are  $\{1, 2, \dots, n\}$  for some  $n$ .*

*Let us  $R_{ij}^k$  be a regular expression such that*

$$L(R_{ij}^k) = \{w \in T^* \mid \delta^{|w|}(i, w) = j, 1 \leq \forall m \leq |w|-1, \delta^m(i, m:w) \leq k\}$$

*The RE  $R_{ij}^k$  denotes the set of strings that take fa  $D$  from state  $i$  to state  $j$  without going through any state numbered higher than  $k$ .*

*When  $k=n$ , no restriction.*

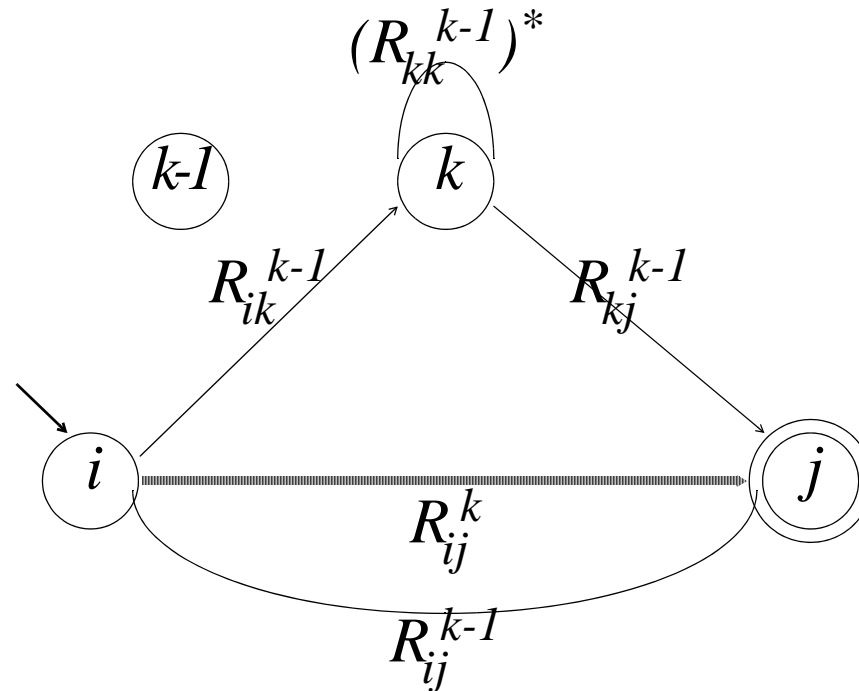
*We can construct  $R_{ij}^k$   $1 \leq \forall i \leq n$ ,  $1 \leq \forall j \leq n$ , and  $0 \leq \forall k \leq n$  by induction on  $k$ .*

**basis:**  *$k = 0$ ,  $1 \leq \forall i \leq n$ ,  $1 \leq \forall j \leq n$ .*

$$\begin{aligned}
 R_{ij}^0 &= \mathbf{a}_1 + \mathbf{a}_2 + \dots + \mathbf{a}_n && \text{if } i \neq j, \text{ and } 1 \leq \forall k \leq n, \delta(i, a_k) = j, \\
 &= \mathbf{a}_1 + \mathbf{a}_2 + \dots + \mathbf{a}_n + \epsilon && \text{if } i = j \text{ and } 1 \leq \forall k \leq n, \delta(i, a_k) = j, \\
 &= \emptyset && \text{otherwise.}
 \end{aligned}$$

**induction:** Assume  $1 \leq \forall i \leq n, 1 \leq \forall j \leq n$ , all  $R_{ij}^{k-1}$ 's are known(I.H.).

$$R_{ij}^k = R_{ik}^{k-1} (R_{kk}^{k-1})^* R_{kj}^{k-1} + R_{ij}^{k-1}.$$



Let  $s$  be the **start** state, and  $F = \{f_1, \dots, f_g\}$  be final states.

$$R = R_{sf_1}^n + R_{sf_2}^n + \dots + R_{sf_g}^n \text{ such that } L(R) = L.$$

**Example 3.5**(p. 95-7) Figure 3.4

$$R_{11}^0 = \varepsilon + \mathbf{1} \quad R_{12}^0 = \mathbf{0} \quad R_{21}^0 = \emptyset \quad R_{22}^0 = \varepsilon + \mathbf{0} + \mathbf{1}$$

$$R_{ij}^k = R_{ik}^{k-1} (R_{kk}^{k-1})^* R_{kj}^{k-1} + R_{ij}^{k-1}$$

$$k=1 \quad R_{ij}^1 = R_{i1}^0 (R_{11}^0)^* R_{1j}^0 + R_{ij}^0.$$

$$\begin{aligned} R_{11}^1 &= R_{11}^0 (R_{11}^0)^* R_{11}^0 + R_{11}^0 = (\varepsilon + \mathbf{1})(\varepsilon + \mathbf{1})^*(\varepsilon + \mathbf{1}) + (\varepsilon + \mathbf{1}) \\ &= (\varepsilon + \mathbf{1})\mathbf{1}^*(\varepsilon + \mathbf{1}) + (\varepsilon + \mathbf{1}) = \mathbf{1}^* + (\varepsilon + \mathbf{1}) = \mathbf{1}^*. \end{aligned}$$

$$R_{12}^1 = R_{11}^0 (R_{11}^0)^* R_{12}^0 + R_{12}^0 = (\varepsilon + \mathbf{1})(\varepsilon + \mathbf{1})^*\mathbf{0} + \mathbf{0} = \mathbf{1}^*\mathbf{0} + \mathbf{0} = \mathbf{1}^*\mathbf{0}.$$

$$R_{21}^1 = R_{21}^0 (R_{11}^0)^* R_{11}^0 + R_{21}^0 = \emptyset(\varepsilon + \mathbf{1})^*(\varepsilon + \mathbf{1}) + \emptyset = \emptyset.$$

$$R_{22}^1 = R_{21}^0 (R_{11}^0)^* R_{12}^0 + R_{22}^0 = \emptyset(\varepsilon + \mathbf{1})^*\mathbf{0} + \varepsilon + \mathbf{0} + \mathbf{1} = \varepsilon + \mathbf{0} + \mathbf{1}.$$

$$k=2 \quad R_{ij}^2 = R_{i2}^1 (R_{22}^1)^* R_{2j}^1 + R_{ij}^1.$$

$$R_{11}^2 = R_{12}^1 (R_{22}^1)^* R_{21}^1 + R_{11}^1 = \mathbf{1}^* \mathbf{0} (\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1})^* \emptyset + \mathbf{1}^* = \mathbf{1}^*.$$

$$\begin{aligned} R_{12}^2 &= R_{12}^1 (R_{22}^1)^* R_{22}^1 + R_{12}^1 \\ &= \mathbf{1}^* \mathbf{0} (\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1})^* (\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1}) + \mathbf{1}^* \mathbf{0} = \mathbf{1}^* \mathbf{0} (\mathbf{0} + \mathbf{1})^*. \end{aligned}$$

$$R_{21}^2 = R_{22}^1 (R_{22}^1)^* R_{21}^1 + R_{21}^1 = (\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1}) (\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1})^* \emptyset + \emptyset = \emptyset.$$

$$\begin{aligned} R_{22}^2 &= R_{22}^1 (R_{22}^1)^* R_{22}^1 + R_{22}^1 \\ &= (\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1}) (\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1})^* (\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1}) + \boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1} = (\mathbf{0} + \mathbf{1})^*. \end{aligned}$$

$$R = R_{12}^2 = \mathbf{1}^* \mathbf{0} (\mathbf{0} + \mathbf{1})^*.$$

### 3.2.2 Converting DFA's to Regular Expressions by Eliminating States

*Previous construction*

$n^3$  equations

$O(4^n)$  symbols in the regular expression

***Eliminating states***

*If we eliminate state  $s$ , all paths that went through  $s$  no longer exist.*

*labels: symbols  $\rightarrow$  possibly infinite strings*

$\rightarrow$  ***regular expression(closure)***

***simultaneous equations( 연립방정식 )***

*n-equations and n-variables*

*eliminating variables*

**Simultaneous equations** for each state with regular expressions

Let  $A = (Q, T, \delta, q_0, F)$  be a FA and  $|Q| = n$ . Then  $\forall q \in Q$ ,  $R_q$  is a **regular equation**

$$1 \leq \forall i \leq n, R_{q_i} = r_{i1}R_{q_1} + r_{i2}R_{q_2} + \dots + r_{in}R_{q_n} + s_i. \quad \text{regular eq.}$$

$$\text{where } 1 \leq \forall j \leq n, \exists m \geq 0, r_{ij} = \mathbf{x}_1 + \mathbf{x}_2 + \dots + \mathbf{x}_m,$$

$$\text{for } 1 \leq \forall k \leq m, q_j \in \delta(q_i, \mathbf{x}_k) \text{ where } \mathbf{x}_k \in \Sigma^* \text{ and}$$

$$s_i = \varepsilon, \text{ if } q_i \in F, \text{ and } s_i = \emptyset, \text{ if } q_i \notin F.$$

Note  $r_{ij}$ 's and  $s_i$ 's are **constant** regular expressions  
and  $R_{q_i}$ 's are unknown **variables**.

Then  **$n$**  states(variables) and  **$n$**  equations.

Note that the **reg. eq.**  $1 \leq \forall i \leq n, R_{q_i} = r_{i1}R_{q_1} + r_{i2}R_{q_2} + \dots + r_{in}R_{q_n} + s_i$   
will be changed to **regular grammar** TBD in Chap 5.



We can solve the **linear simultaneous equation**

1. eliminate **variables(states)** by **substitution**( 대입 )
2. eliminate **recursive variable** by **closure**.

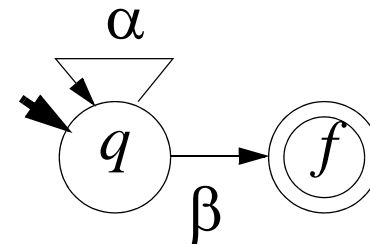
Let  $q = \alpha q + \beta$  where  $\alpha$  and  $\beta$  are **regular equation with variables**.

$$q \Rightarrow^{q=\beta} \beta \text{ or}$$

$$q \Rightarrow^{q=\alpha q} \alpha q \Rightarrow^{q=\beta} \alpha \beta \text{ or}$$

$$q \Rightarrow^{q=\alpha q} \alpha q \Rightarrow^{q=\alpha q} \alpha \alpha q \Rightarrow^{q=\beta} \alpha \alpha \beta \text{ or}$$

...



$$q \Rightarrow^{(q=\alpha q)^n} \alpha^n q \Rightarrow^{q=\beta} \beta \quad n \geq 0.$$

$$q = \alpha^* \beta.$$

**Example 3.5 revisited**(p. 95) Fig. 3.4(p. 96)

$$A = 1A + 0B \qquad A = 1A + 0(0+1)^* = 1^*0(0+1)^* \quad \alpha^*\beta.$$

$$B = (0+1)B + \varepsilon \qquad B = (0+1)^*\varepsilon = (0+1)^* \quad \alpha^*\beta.$$

**Example 3.6** Figure 3.12(pp 101)

*regular grammar*

$$A = (0+1)A + 1B$$

$$A \rightarrow 0A \mid 1A \mid 1B$$

$$B = (0+1)C$$

$$B \rightarrow 0C \mid 1C$$

$$C = (0+1)D + \varepsilon$$

$$C \rightarrow 0D \mid 1D \mid \varepsilon$$

$$D = \varepsilon$$

$$D \rightarrow \varepsilon$$

$$C = (0+1)D + \varepsilon = (0+1)\varepsilon + \varepsilon = 0+1 + \varepsilon$$

$$B = (0+1)C = (0+1)(0+1 + \varepsilon) = (0+1)^2 + (0+1) = (0+1)(0+1 + \varepsilon)$$

$$A = (0+1)A + 1B = (0+1)A + 1(0+1)(0+1 + \varepsilon)$$

$$= (0+1)^*1(0+1)(0+1 + \varepsilon) \quad \alpha^*\beta.$$

**NFA** Figure 2.9(pp 56)

$$C = \varepsilon \qquad B = 1C = 1\varepsilon = 1$$

$$A = (0 + 1)A + 0B = (0 + 1)A + 01 = (0 + 1)^* 01$$

$$\therefore (1 + 0(0 + 10)^* 11)^* 0(0 + 10)^* 1 = (0 + 1)^* 01.$$

DFA in Figure 2.14 (pp 63) revisited

$$A = 1A + 0B$$

$$B = 0B + 1C$$

$$C = 1A + 0B + \varepsilon$$

Solution 1

$$B = 0B + 11A + 10B + 1 = (0+10)B + 11A + 1 = (0+10)^* (11A + 1)$$

$$A = 1A + 0(0+10)^* (11A + 1) = (1 + 0(0+10)^* 11)A + 0(0 + 10)^* 1$$

$$= (1 + 0(0 + 10)^* 11)^* 0(0 + 10)^* 1.$$

Solution 2

$$C = \underline{0B} + \underline{1A} + \varepsilon = A + \varepsilon \quad B = 0B + 1C = \underline{0B} + \underline{1A} + 1\varepsilon = A + 1$$

$$A = \underline{0B} + \underline{1A} = 1A + 0A + 01 = (0 + 1)A + 01 = (0 + 1)^* 01.$$

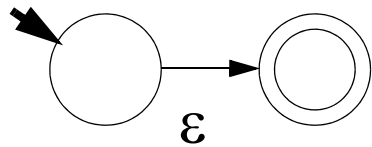
### 3.2.3 Converting Regular Expressions to Automata

**Theorem 3.7** Every language defined by a regular expression is also defined by a finite automaton.

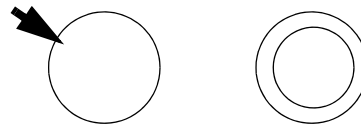
**Proof** Suppose  $L = L(R)$  for some regular expression  $R$ .

We show that  $L = L(E)$  for some  $\varepsilon$ -NFA  $E$ .

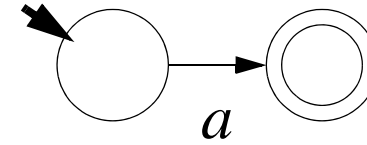
**basis:**



1.  $\varepsilon$



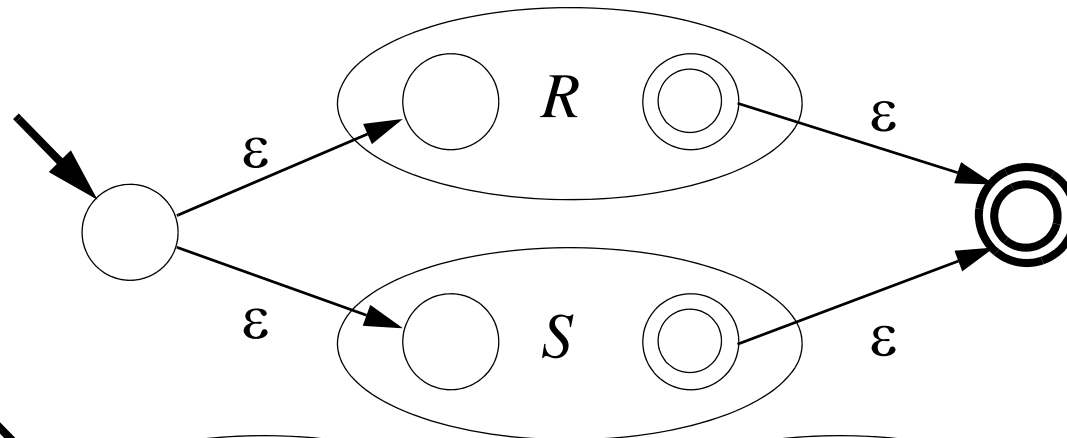
2.  $\emptyset$



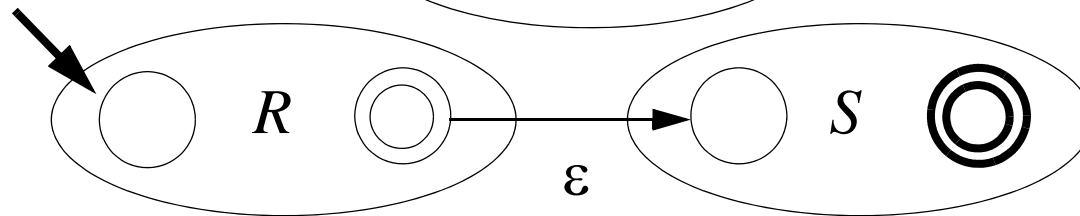
3. **a**

*induction:*

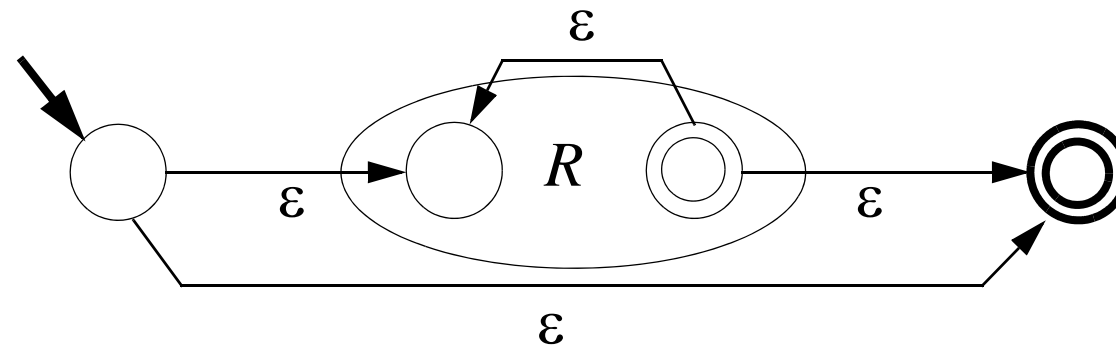
1.  $R + S$



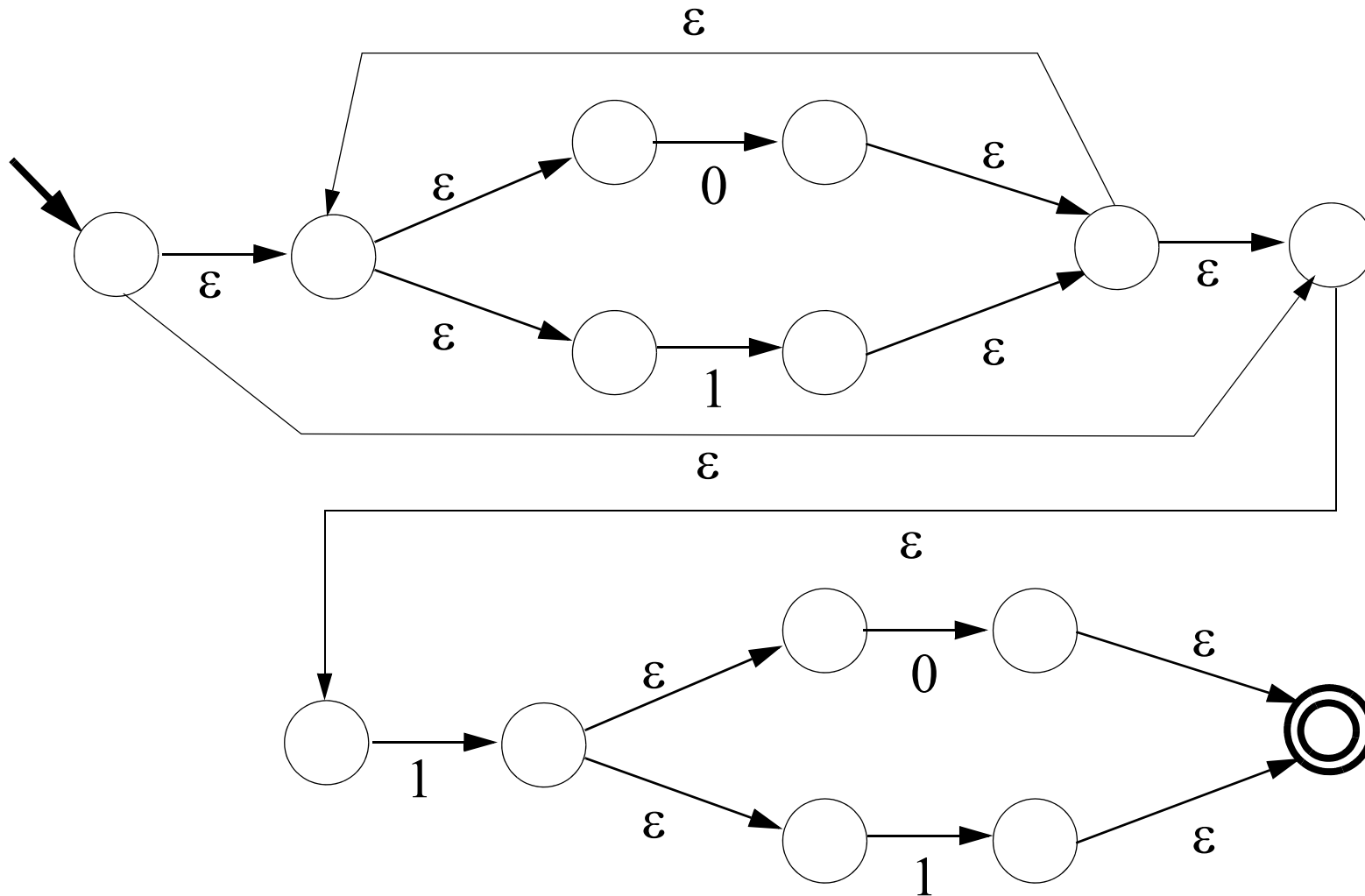
2.  $RS$



3.  $R^*$



**Example 3.8 and Fig. 3.18**  $\epsilon$ -NFA for RE  $(0+1)^*1(0+1)$  in p. 106.



### 3.4 Algebraic Laws for Regular Expressions

Let  $R, S, T$  be regular expressions over  $\Sigma$ .

$$R + S = S + R$$

*union is commutative*

$$(R + S) + T = R + (S + T)$$

*union is associative*

$$RS \neq SR$$

*concatenation is non-commutative*

$$(RS)T = R(ST)$$

*concatenation is associative*

$$\emptyset + R = R + \emptyset = R$$

*$\emptyset$  is the identity for union*

$$\varepsilon R = R\varepsilon = R$$

*$\varepsilon$  is the identity for concatenation*

$$\emptyset R = R\emptyset = \emptyset$$

*$\emptyset$  is the annihilator for concatenation*

$$R(S + T) = RS + RT$$

*concatenation distributes over union*

$$(S + T)R = SR + TR$$

$$R + R = R$$

$$(R^*)^* = R^*$$

*Union is idempotent*

*Closure is idempotent*

$$\emptyset^* = \varepsilon \quad \varepsilon^* = \varepsilon$$

*but*  $\emptyset^+ = \emptyset \quad \varepsilon^+ = \varepsilon.$

$$R^+ = RR^* = R^*R.$$

$$R^* = R^+ + \varepsilon$$

*but*  $R^+ \neq R^* - \{\varepsilon\}, \text{ if } \varepsilon \in R.$

$$T^* = T^+ + \varepsilon$$

*and*  $T^+ = T^* - \{\varepsilon\} \text{ (since } \varepsilon \notin T).$

***Theorem 3.A Any finite language is regular.***

***Proof Any finite language can be denoted by (finite) regular expression.***

***union and concatenation.***

***no closure***

***finite***



*Following statements are logically equivalent*

1.  $L$  is **regular**.
2.  $L = L(D)$  for some DFA  $D$  with total function  $\delta$ .
3.  $L = L(P)$  for some DFA  $P$  with partial function  $\delta$ .
4.  $L = L(N)$  for some NFA  $N$ .
5.  $L = L(E)$  for some  $\varepsilon$ -NFA  $E$ .
6.  $L = L(X)$  for some XFA  $X$ .
7.  $L = L(R)$  for some RE  $R$ .

*Following statements are logically equivalent*

1.  $L$  is **regular**.
2.  $L = L(A)$  for some **finite automaton**  $A$ .
3.  $L = L(R)$  for some **regular expression**  $R$ .
4.  $L = L(G)$  for some **regular grammar**  $G$ . (TBD in Chap. 5)

**Chomsky's type 3 Languages = Regular Languages = Finite Automata  
 = Regular Expressions = Regular Grammars (Chap. 5)**

