Chap. 3 Regular Expressions and Languages

3.1.2 Regular Expressions *over* ***some alphabet*** *T.*

Basis*:*

1. The ***constant*** ε *is a* ***regular expression****,*
denoting the languages $\{\varepsilon\}$*, i.e.,* $L(\varepsilon) = \{\varepsilon\}$.

2. The ***constant*** $\varnothing$ *is a* ***regular expression****,*
denoting the languages $\varnothing$*, i.e.,* $L(\varnothing) = \varnothing$.

3. If $a \in T$*, then* **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 justaxaposed.
3. union(+)

Example 3.3 in p.91

Equivalence of regular expresssions

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, …, n} for some n.*

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

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

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*:* $k = 0,\ 1 \leq {}^{\forall}i \leq n,\ 1 \leq {}^{\forall}j \leq n.$

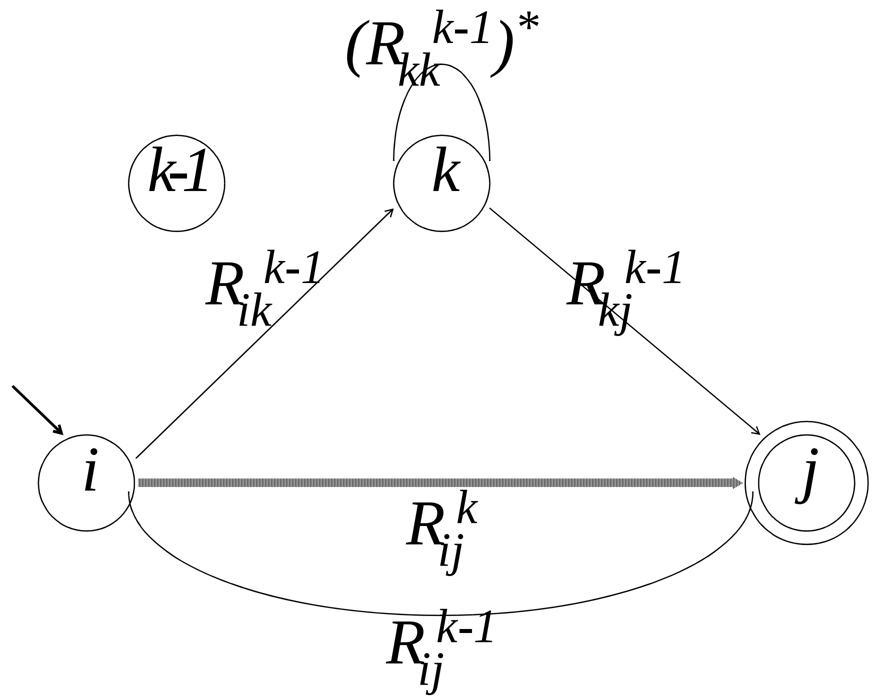
$R_{ij}^{0} = \mathbf{a}_1 + \mathbf{a}_2 + \ldots + \mathbf{a}_n$ *if* $i \neq j$, *and* $1 \leq \forall k \leq n$, $\delta(i, a_k) = j$,

$\quad\;\; = \mathbf{a}_1 + \mathbf{a}_2 + \ldots + \mathbf{a}_n + \varepsilon$ *if* $i = j$ *and* $1 \leq \forall k \leq n$, $\delta(i, a_k) = j$,

$\quad\;\; = \varnothing$ *otherwise.*

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* $F = \{f_1, \ldots, f_g\}$ *be final states.*

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

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

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

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

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

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

$\qquad = (\varepsilon + \mathbf{1})\mathbf{1}^*(\varepsilon + \mathbf{1}) + (\varepsilon + \mathbf{1}) = \mathbf{1}^* + (\varepsilon + \mathbf{1}) = \mathbf{1}^*.$

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

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

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

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

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

$R_{12}^{2} = R_{12}^{1} (R_{22}^{1})^{*} R_{22}^{1} + R_{12}^{1}$

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

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

$R_{22}^{2} = R_{22}^{1} (R_{22}^{1})^{*} R_{22}^{1} + R_{22}^{1}$

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

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

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 though s no longer exists.

*labels: symbols → possibly **infinite** strings*

*→ **regular expression(closure)***

***simultaneous** equations(연립방정식)*

n-equations and n-variables

eliminating variables

Simultaneous equations *for each state with regular expressions*

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

$$1 \le {}^\forall i \le n,\ R_{q_i} = r_{i1}R_{q_1} + r_{i2}R_{q_2} + \dots + r_{in}R_{q_n} + s_i. \qquad \textbf{regular eq.}$$

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

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

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

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.*** $1 \le {}^\forall i \le n,\ R_{q_i} = r_{i1}R_{q_1} + r_{i2}R_{q_2} + \dots + r_{in}R_{q_n} + s_i,$

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* α *and* β *are **regular equation** with **variables**.*

$q \Rightarrow \beta$ *or*

$q \Rightarrow \alpha + \beta$ *or*

$q \Rightarrow \alpha\alpha + \beta = \alpha^2 + \beta$ *or*

...

$q = \alpha^* \beta$.

strange solution

$q \neq 1/(1-\alpha)\, \beta$

$= (1 + \alpha + \alpha^2 + \ldots)\, \beta$

$\neq \alpha^* \beta$.

Example 3.6 Figure 3.12(pp 101)

$A = (\mathbf{0} + \mathbf{1})A + \mathbf{1}B$
$B = (\mathbf{0} + \mathbf{1})C$
$C = (\mathbf{0} + \mathbf{1})D + \varepsilon$
$D = \varepsilon$

regular grammar

$A \to 0A \mid 1A \mid 1B$
$B \to 0C \mid 1C$
$C \to 0D \mid 1D \mid \varepsilon$
$D \to \varepsilon$

$$C = (\mathbf{0} + \mathbf{1})D + \varepsilon = (\mathbf{0} + \mathbf{1})\varepsilon + \varepsilon = \mathbf{0} + \mathbf{1} + \varepsilon$$

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

$$A = (\mathbf{0} + \mathbf{1})A + \mathbf{1}B = (\mathbf{0} + \mathbf{1})A + \mathbf{1}(\mathbf{0} + \mathbf{1})(\mathbf{0} + \mathbf{1} + \varepsilon)$$

$$= (\mathbf{0} + \mathbf{1})^*\mathbf{1}((\mathbf{0} + \mathbf{1})(\mathbf{0} + \mathbf{1} + \varepsilon)) \qquad \alpha^*\beta.$$

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

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

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

DFA in Figure 2.14 (pp 63) revisited

$A = \mathbf{1}A + \mathbf{0}B \qquad A = \mathbf{1}A + \mathbf{0}(\mathbf{0} + \mathbf{10})^*(\mathbf{11}A + \mathbf{1})$

$= (\mathbf{1} + \mathbf{0}(\mathbf{0} + \mathbf{10})^*\mathbf{11})A + \mathbf{0}(\mathbf{0} + \mathbf{10})^*\mathbf{1}$

$= (\mathbf{1} + \mathbf{0}(\mathbf{0} + \mathbf{10})^*\mathbf{11})^*\mathbf{0}(\mathbf{0} + \mathbf{10})^*\mathbf{1}.$

$B = \mathbf{0}B + \mathbf{1}C \qquad B = \mathbf{0}B + \mathbf{11}A + \mathbf{10}B + \mathbf{1} = (\mathbf{0} + \mathbf{10})B + \mathbf{11}A + \mathbf{1}$

$= (\mathbf{0} + \mathbf{10})^*(\mathbf{11}A + \mathbf{1})$

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

NFA Figure 2.9(pp 56)

$C = \varepsilon \qquad B = \mathbf{1}C = \mathbf{1}\varepsilon = \mathbf{1}$

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

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

DFA in Figure 2.14 (pp 63) revisited

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

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

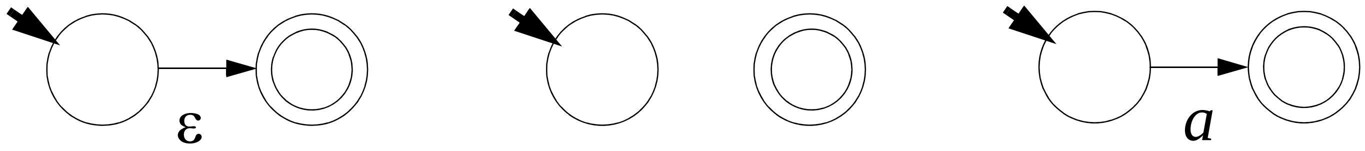
3.2.3 Converting Regular Expressions to Automata

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 ε-NFA E.

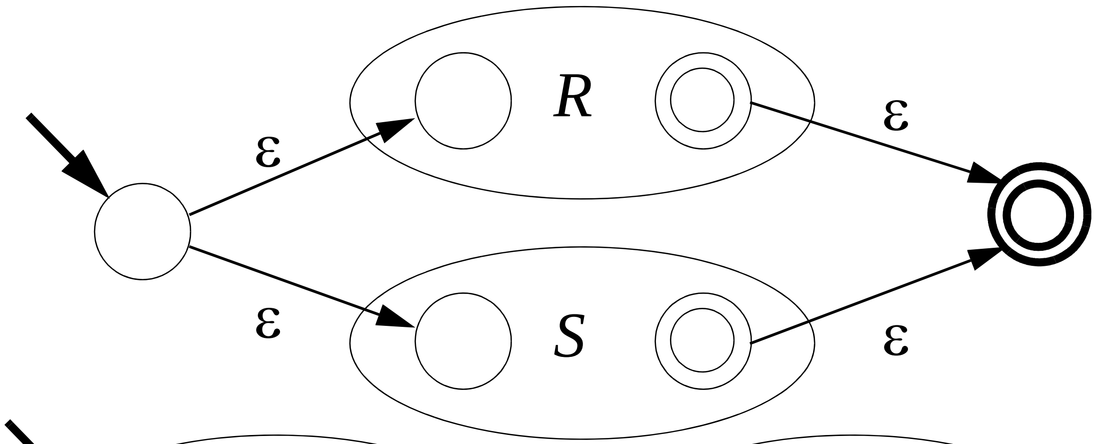
basis*:*



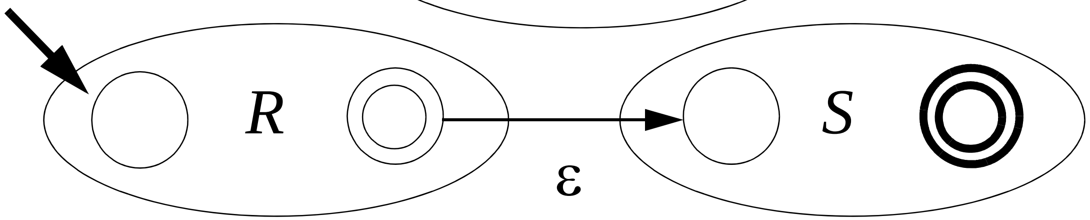
1. ε *2.* ∅ *3.* **a**

induction:

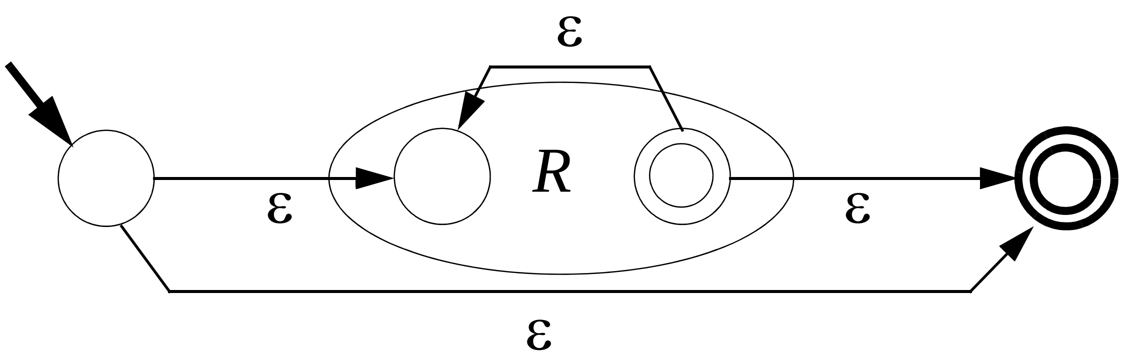
1. R + S



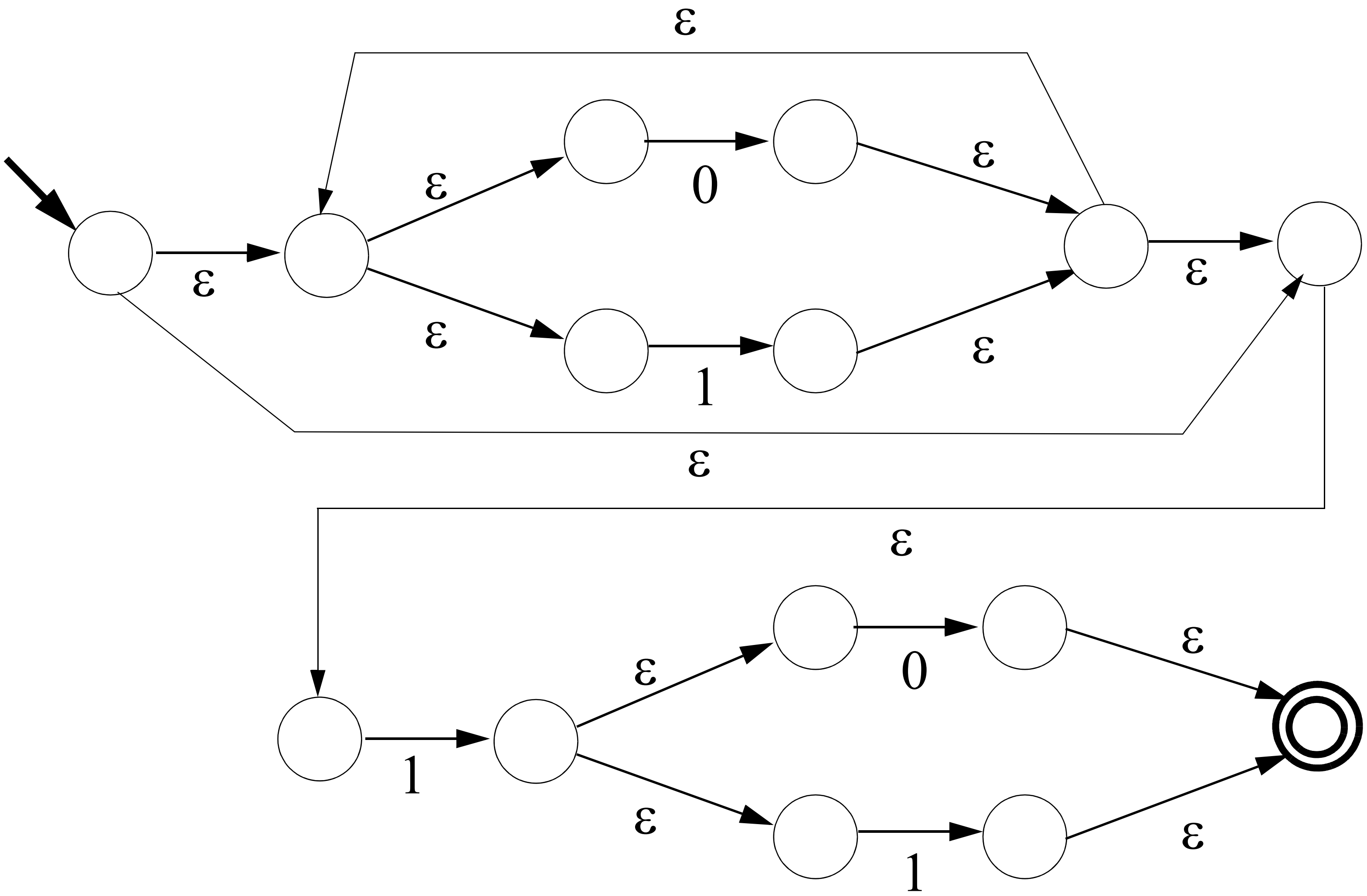
2. R S



3. R^*



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



3.4 Algebraic Laws for Regular Expressions

Let R, S, T be regular expressions over Σ.

$R + S = S + R$	***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***
$\varnothing + R = R + \varnothing = R$	$\varnothing$ *is the* ***identity*** *for* ***union***
$\varepsilon R = R\varepsilon = R$	ε *is the* ***identity*** *for* ***concatenation***
$\varnothing R = R\varnothing = \varnothing$	$\varnothing$ *is the* ***annihilator*** *for* ***concatenation***
$R(S + T) = RS + RT$	***concatenation distributes*** *over* ***union***
$(S + T)R = SR + TR$	

$R + R = R$ — Union is ***idempotent***

$(R^*)^* = R^*$ — Closure is ***idempotent***

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

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

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

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

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 δ.
3. L = L(P) for some DFA P with partial function δ.
4. L = L(N) for some NFA N.
5. L = L(E) for some ε-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)