Definition: given an alphabet \( \Sigma \), regular expressions are strings over the alphabet \( \Sigma \cup \{ +, \times, (, ), \cdot, \epsilon, \phi \} \) defined inductively as follows:

- **basis**: \( \epsilon, \phi \), and each \( a \in \Sigma \) is a R.E.
- **inductive step**: if \( E \) and \( F \) are R.E. then so are:
  - \( E + F \) (union)
  - \( E \cdot F \) (concatenation)
  - \( E^* \) (closure)
  - \( (E) \) (parentheses)

Example: 
\[
\epsilon \cdot (a + b)^* \cdot b^* \cdot \epsilon
\]

Definition: language \( L(E) \) defined by a R.E. \( E \)
in also defined inductively:

- \( L(\epsilon) = \{ \epsilon \} \) empty word
- \( L(\phi) = \emptyset \) empty language
- \( L(a) = \{ a \} \)
- \( L(E + F) = L(E) \cup L(F) \)
- \( L((E)) = L(E) \)
- \( L(E \cdot F) = L(E) \cdot L(F) \) ... concatenation

concatenation of two languages \( L_1 \) and \( L_2 \):

\[
L_1 \cdot L_2 = \{ \omega | \omega = \xi \cdot \eta, \xi \in L_1, \eta \in L_2 \}
\]

Example: 
\[
E = \epsilon + 1 \quad \Rightarrow \quad L(E) = \{ \epsilon, 1 \}
\]
\[
F = \epsilon \cdot 0 + 1 \quad \Rightarrow \quad L(F) = \{ \epsilon, 0, 1 \}
\]
\[
G = E \cdot F \quad \Rightarrow \quad L(G) = \{ \epsilon, 0, 1, 10, 11 \}
\]
\[ L(E^*) = (L(E))^* \quad \text{... closure} \]

### Closure of a language L?

We first define the powers of a language L:

- \( L^0 = \{ \varepsilon \} \)
- \( L^k = L^{k-1} \cdot L \)

Hence \( L^k = \{ w \mid w = x_1 \cdots x_k \text{, with } \forall x_i, x_n \in L \} \)

Closure of L: \( L^* = L^0 \cup L^1 \cup L^2 \cup \cdots \)

**Example:**

\[ \begin{align*}
E = 0 + 1 & \quad \Rightarrow L(E) = \{0, 1\} \\
F = E^* & \quad \Rightarrow L(F) = \text{set of all binary strings} \\
E = 0 \cdot 0 & \quad \Rightarrow L(E^*) = \{ \epsilon, 00, 0000, 000000, \ldots \}
\end{align*} \]

= all even-length strings of 0's

---

**Positive closure of a language L**

\[ L^+ = L^0 \cup L^1 \cup \ldots \]

We can introduce a positive closure operator on R.E.

\[ L(E^+) = (L(E))^+ \]

---

**Note:** we have to distinguish between an expression E and the language \( L(E) \) defined by E.

When we write \( E = F \), we usually mean not syntactic equality, but equality of the corresponding languages, i.e., \( L(E) = L(F) \).

In other words, equality is in the alphabet of R.E.

---

**Precedence of operators:**

- High: \( \ast \)
- Low: \( + \)

**Example:** \( E + F \cdot G^* = E + (F \cdot (G^*)) \)
Algebraic Laws for R.E.

Similar to the laws for arithmetic expressions, we can express laws for R.E.: treat * as sum and + as product.

- Associativity of * and +
  
  \[(E \cdot F) \cdot G = E \cdot (F \cdot G) = E \cdot F \cdot G\]
  
  \[(E + F) + G = E + (F + G) = E + F + G\]

- Commutativity of +
  
  \[E + F = F + E\]

  Note: * is not commutative: \[E \cdot F \neq F \cdot E\]

- Distributivity:
  
  1) Left distributive law of * over +: \[E \cdot (F + G) = E \cdot F + E \cdot G\]

  2) Right: \[\quad (F + G) \cdot E = F \cdot E + G \cdot E\]

  Proof of 1: The law actually holds for arbitrary languages, and does not require \(E, F, G\) to be R.E.

  Hence, we prove: for arbitrary languages \(L, M, N\):

  \[L \cdot (M \cup N) = L \cdot M \cup L \cdot N\]

  We show that for a string \(w\) we have \(w \in L \cdot (M \cup N)\) if and only if \(w \in L \cdot M \cup L \cdot N\)

  "Only if": \(w \in L \cdot (M \cup N) \Rightarrow w = x \cdot y\) with \(x \in L, y \in M \cup N\)

  Since \(y \in M \cup N\), either \(y \in M\) or \(y \in N\) (or both)

  5) If \(y \in M\), then \(w = x \cdot y \in L \cdot M\), hence \(w \in L \cdot M \cup L \cdot N\)

     (Similarly for \(y \in N\))

  "If": \(w \in L \cdot M \cup L \cdot N\), hence either \(w \in L \cdot M\) or \(w \in L \cdot N\).

  If \(w \in L \cdot M\), then \(w = x \cdot y\), with \(x \in L, y \in M\).

  (Similarly, hence \(y \in M \cup N\) and \(w = x \cdot y \in L \cdot (M \cup N)\). For \(w \in L \cdot N\)
Example: \( 0 \cdot 0 + 0 \cdot 1^* = 0 \cdot (0 + 1^*) \)

we can factor out \( e \in 0 \) from the union

What about \( 0 + 0 \cdot 1^* \)?

if we factor out \( e \in 0 \), what remains after the summation

on the left?

\[ 0 + 0 \cdot 1^* = 0 \cdot E + 0 \cdot 1^* = 0 \cdot (E + 1^*) = 0 \cdot 1^* \]

\[ \text{idem} \quad \text{since } E \in L(1^*) \]

- Identities and annihilators (hold for arbitrary languages)
  - \( \emptyset + E = E + \emptyset = E \)
  - \( \varepsilon \cdot E = E \cdot \varepsilon = E \)
  - \( \emptyset \cdot E = E \cdot \emptyset = \emptyset \)

- Idempotency
  - \( E + E = E \)
  - \( (E^*)^* = E^* \)
  - \( \text{Proof: Exercise 3.4.1} \)

- Other laws for closure (already seen)
  - \( \emptyset^* = \varepsilon \)
  - \( \varepsilon^* = \varepsilon \)
  - \( E^+ = E \cdot E^* = E^* \cdot E \)
  - \( E^* = E^+ + \varepsilon \)

Note: if \( E \in L(E) \), then \( E^* = E^+ \)

Exercise 3.4.4: Prove that \( (E^* F^*)^* = (E + F)^* \)
Exercise 3.1.1 Write R.E.'s for the following languages

a) \( \{ w \in \{e, b, c\}^* \mid w \text{ contains at least one } e \text{ and at least one } b \} \)

b) \( \{ w \in \{0, 1\}^* \mid w \text{'s tenth symbol from the right is a 1} \} \)

c) \( \{ w \in \{0, 1\}^* \mid w \text{ contains at most one pair of consecutive 1's} \} \)

Exercise 3.1.2 Write R.E.'s for the following languages

a) The set of all strings over \( \{0, 1\} \) s.t. every pair of adjacent 0's appears before any pair of adjacent 1's

b) The set of strings of 0's and 1's whose number of 0's is divisible by 5

Solutions:

3.1.1 a) \( (c^*a^*(e+c)^*b(e+b+c)^*)^* + c^*b^*(b+c)^*a^*(e+b+c)^* \)

b) \( (0+1)^* \underbrace{1\cdot(0+1)\cdot\ldots\cdot(0+1)}_{3 \text{ times}} \)

c) \( 0^* (1.0^+)^* \cdot 1.1. \cdot (0^+).0^* + 0^* (1.0^+)^* \)

3.1.2 a) \( \underbrace{0^* (1.0^+)^* \cdot 1^* (0.1^+)^*}_{\text{no pair of adjacent 1's}} \)

b) \( \underbrace{1^* .0^* .1^* .0^* .1^* .0^* .1^* .0^* .1^*}_{} \)
What is the relationship between the classes of languages studied so far?

\[ \text{\textit{\varepsilon}-elimination} \quad \text{\textit{\varepsilon}-subset-construction} \]

\[ \varepsilon\text{-NFA} \leftrightarrow \text{NFA} \leftrightarrow \text{DFA} \]

\[ \text{regular languages} \quad \mapsto \quad \text{R.E.} \quad \mapsto \quad ? \]

**Theorem:** (R.E. $\rightarrow$ $\varepsilon$-NFA)

For every R.E. $E$ there is an $\varepsilon$-NFA $A_E$ such that $L(A_E) = L(E)$.

**Proof:** Let us call an $\varepsilon$-NFA simple if

- it has only one initial state
- the initial state has no incoming arcs
- the final state has no outgoing arcs

We show by structural induction that for each R.E. $E$ there is a simple $\varepsilon$-NFA $A_E$ such that $L(E) = L(A_E)$.

**Basis:** $E = \varepsilon$, $E = \emptyset$, $E = e$ for some $e \in \Sigma$

\[ A_\varepsilon \rightarrow 90 \rightarrow \varepsilon \rightarrow 96 \]

\[ A_\emptyset \rightarrow 90 \rightarrow 96 \]

\[ A_e \rightarrow 90 \rightarrow e \rightarrow 96 \]

**Inductive case:**
1. $E = F \cup G$
2. $E = F \cdot G$
3. $E = F^*$
4. $E = (F)$
By I.H., there are simple $\varepsilon$-NFA's for $F$ and $G$.

1) $E = F + G$

$L(A_E) = L(A_F) \cup L(A_G) = L(F) \cup L(G) = L(F + G) = L(E)$

by I.H.

2) $E = F \cdot G$

$L(A_E) = L(A_F) \cdot L(A_G) = L(F) \cdot L(G) = L(F \cdot G) = L(E)$

by I.H.

3) $E = F^*$

$L(A_E) = L(F)^* = L(E)$

by I.H.

3) $E = (F)$

$A_E = A_F$

q.e.d.
Example: \( E = Q^* + b \cdot c \)

```
Example: E = Q^* + b \cdot c
```

**Theorem (DFA \( \rightarrow \) R.E.)**

For every DFA \( A \) there is a R.E. \( E_A \) s.t. \( L(E_A) = L(A) \)

**Proof:** Set \( A = (Q, \Sigma, \delta, q_0, F) \)

We assume without loss of generality (W.L.O.G.) that \( Q = \{ q_1, q_2, \ldots, q_n \} \)

Let \( n \) define \( L_{ij} = \{ w \mid \hat{\delta}(q_i, w) = q_j \} = \{ w \mid \text{w takes A from } q_i \text{ to } q_j \} \)

Note that \( L_{ij} \) is \( L(A_{ij}) \) with \( A_{ij} = (Q, \Sigma, \delta, q_i, \{ q_j \}) \)

We aim at constructing R.E.s \( E_{ij} \) for \( L_{ij} \).

Then we can take \( E_A = \sum_{q_0 \in F} E_{q_0} \), since

\[
L(E_A) = U_{q_0 \in F} L(E_{q_0}) = U_{q_0 \in F} \{ w \mid \hat{\delta}(q_i, w) \in F \} = L(A)
\]

How can we compute \( E_{ij} \)?

Let us define \( \forall i, j \in \{ 1, \ldots, n \} \), \( \forall k \in \{ 0, \ldots, n \} \)

\[
L_{ij}^k = \{ w \mid \text{A goes from } q_i \text{ to } q_j \text{ on input } w, \\
\text{passing only through } q_1, \ldots, q_k \text{ as intermediate states} \}
\]
Example:

\[ \begin{array}{c}
q_1 \xrightarrow{a} q_2 \xrightarrow{b} q_3 \xrightarrow{c} q_4 \xrightarrow{d} q_5 \xrightarrow{f} q_6 \\
q_7 \xrightarrow{e} q_8 \\
\end{array} \]

\[ \begin{align*}
abc & \in L_{15}^3 \\
def & \notin L_{15}^3, \quad \text{but}\quad def \in L_{15}^4 \\
def & \in L_{15} \\
\end{align*} \]

\[ L_{12} = \{e, d\} \]

\[ L_{15} = \{abc, dbc\} \]

\[ L_{15}^3 = L_{15}^5 = L_{15} = \{abc, dbc, def, def\} \]

Note: \[ L_{ij}^k = L_{ij} \]

Since we are done if we can construct RE's \[ E_{ij}^k \] for \[ L_{ij}^k \].

We can simply take \[ E_{ij} = E_{ij}^1 \], and hence \[ E_A = \sum_{q \in F} E_{ij}^m \].

We construct \[ E_{ij}^k \] by induction on \[ k \]:

**Base:** we construct \[ E_{ij}^0 \] for all \( i, j \in \{1, \ldots, n\} \).

Since \( k=0 \), we cannot go through any intermediate state.

2 cases: each with 2 sub-cases:

1. \( i \neq j \):
   - \[ q_i \xrightarrow{a} q_j \]
   - \[ E_{ij}^0 = q_i \]
   - \[ E_{ij}^0 = \phi \]

2. \( i = j \):
   - \[ q_i \xrightarrow{e_k} q_i \]
   - \[ E_{ii}^0 = q_i + e_k + \cdots + e_n \]
   - \[ E_{ii}^0 = \varepsilon \]

\[ E_{ii}^0 = \varepsilon \]
Induction: assume we have constructed $E_{ij}^{k-1}$ \( \forall i, j \in \{1, \ldots, n\} \).

We show how to construct $E_{ij}^k$.

Observe:

- $L_{ij}^k$ will include $L_{ij}^{k-1}$.
- It additionally will contain those walks that lead through $q_k$ at least once, when going from $q_i$ to $q_j$.

\[ q_i \xrightarrow{x_1} q_k \xrightarrow{x_2} q_k \xrightarrow{x_3} \ldots \xrightarrow{x_0} q_j \]

\[ w = x_1 \cdot x_2 \cdot \ldots \cdot x_0 \]

where $\rightarrow$ represents transitions going at most through $\{x_1, \ldots, x_{k-1}\}$.

Then $x_1 \in L_{ik}^{k-1}$, $x_2, \ldots, x_{k-1} \in L_{kk}$, $x_0 \in L_{kj}$.

\[ \Rightarrow w \in L_{ik}^{k-1} \cdot \underbrace{L_{kk} \cdot \ldots \cdot L_{kk}}_{(k-1) \text{ times}} \cdot L_{kj} \]

\[ \Rightarrow E_{ij}^k = E_{ij}^{k-1} + \underbrace{E_{ik}^{k-1} \cdot (E_{kk}^{k-1})^* \cdot E_{kj}^{k-1}}_{(k-1) \text{ times}} \]

Example:

\[ q_1 \xrightarrow{\lambda} q_2 \xrightarrow{E_{q_1}^{q_2}} q_3 \]

<table>
<thead>
<tr>
<th>$k$</th>
<th>$E_{11}^k$</th>
<th>$E_{12}^k$</th>
<th>$E_{21}^k$</th>
<th>$E_{22}^k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\varepsilon + 0$</td>
<td>1</td>
<td>(\emptyset)</td>
<td>$\varepsilon + 0 + 1$</td>
</tr>
<tr>
<td>1</td>
<td>$(\varepsilon + 0) + (\varepsilon + 0)^* \cdot (\varepsilon + 0)$</td>
<td>$\varepsilon + 0 + 1$</td>
<td>(\emptyset)</td>
<td>$\varepsilon + 0 + 1$</td>
</tr>
<tr>
<td>2</td>
<td>not needed</td>
<td>$\varepsilon + 0 + (\varepsilon + 0 + 1)^*$</td>
<td>not needed</td>
<td>not needed</td>
</tr>
</tbody>
</table>

$E_{11}^2 = E_{12}^2 = \varepsilon + 0 + (\varepsilon + 0 + 1)^*$ is **optional**.
Theorem \((\text{DFA} \rightarrow \text{R.E.})\)

For every DFA \(A\), there is a R.E. \(E_A\) s.t. \(L(E_A) = L(A)\).

**Proof Sketch:** We show how to construct \(E_A\) by eliminating states of \(A\).

Consider the elimination of a state \(s_i\):

- If there was a path from state \(p\) to state \(q\) over \(s_i\),
  - after eliminating \(s_i\) the path does no longer exist.
  - we have to compensate for that.

We add a regular expression "connecting" \(p\) and \(q\) and capturing the missing path.

- We can eliminate in this way all states except initial and final states.

**Strategy:**

a) For each final state \(q\), eliminate all states except \(q, q_0\).

b) If \(q \neq q_0\), we are left with:

\[
(R + S \cdot U^* \cdot T)^* \cdot S \cdot U^*
\]

- If \(q = q_0\), we must eliminate all states except \(q_0\).

We are left with \(R^*\).

- We take the union of all derived R.E.s.
We view all edge labels as R.E.'s (missing labels mean $\phi$).

Eliminate $B$:

$$E = \varnothing \cup 1 \cdot \varnothing^* \cdot (0+1) = 1 \cdot \varnothing^* \cdot (0+1) = 1 \cdot (0+1)$$

Eliminate $C$:

$$E_1 = (0+1)^* \cdot E = (0+1)^* \cdot 1 \cdot (0+1) \cdot (0+1)$$

$$E_2 = (0+1)^* \cdot 1 \cdot (0+1)$$

$$E = E_1 \cup E_2 = (0+1)^* \cdot 1 \cdot (0+1) \cdot (0+1) + (0+1)^* \cdot 1 \cdot (0+1)$$

$$= (0+1)^* \cdot 1 \cdot (0+1) \cdot (3 \cdot 0+1)$$