CSE-233 : Section A

Summer 2020

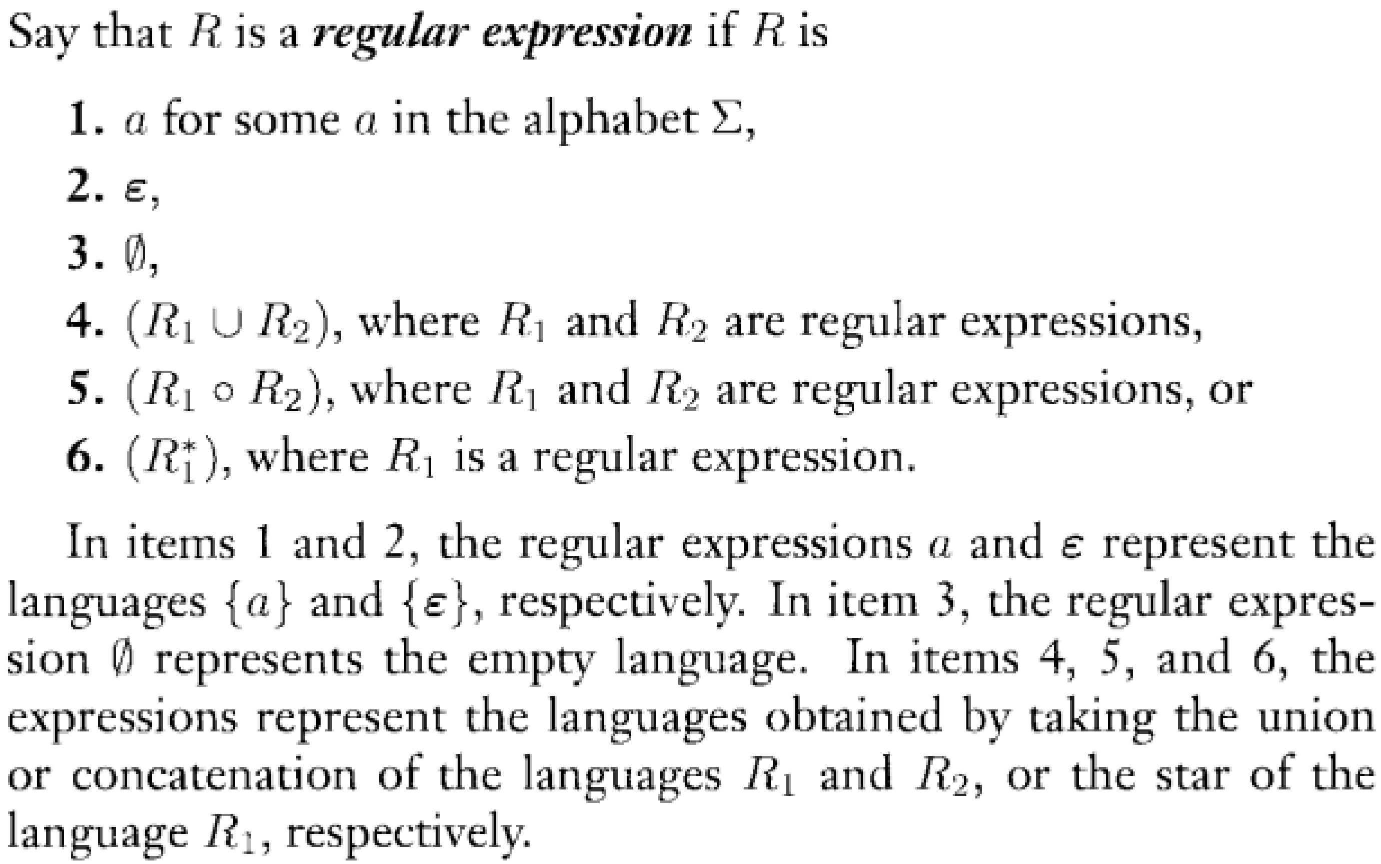
Regular Expressions

Reference:

Book2 Chapter 1.3

Md. Saidul Hoque Anik
anik@cse.uiu.ac.bd

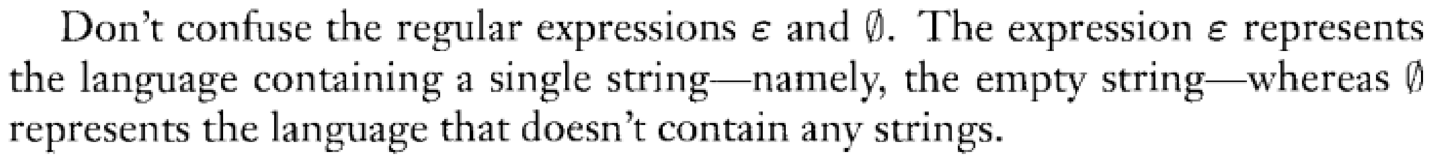
Formal Definition of Regular Expression



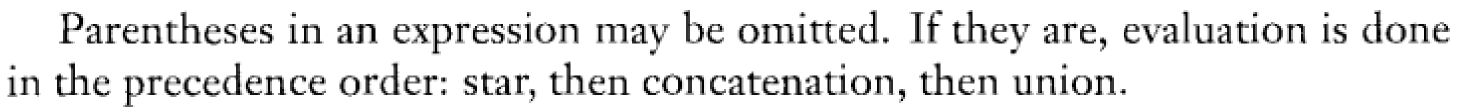
Say that R is a ***regular expression*** if R is

1. a for some a in the alphabet Σ,
2. ε,
3. $\emptyset$,
4. $(R_1 \cup R_2)$, where R_1 and R_2 are regular expressions,
5. $(R_1 \circ R_2)$, where R_1 and R_2 are regular expressions, or
6. (R_1^*), where R_1 is a regular expression.

In items 1 and 2, the regular expressions a and ε represent the languages $\{a\}$ and $\{\varepsilon\}$, respectively. In item 3, the regular expression $\emptyset$ represents the empty language. In items 4, 5, and 6, the expressions represent the languages obtained by taking the union or concatenation of the languages R_1 and R_2, or the star of the language R_1, respectively.

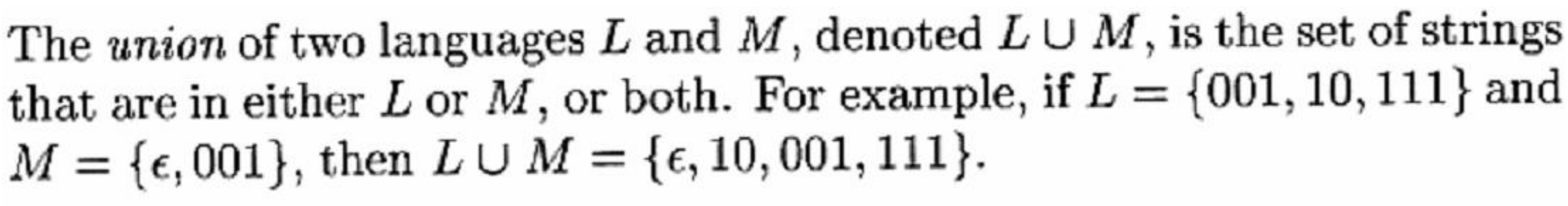


Don't confuse the regular expressions ε and $\emptyset$. The expression ε represents the language containing a single string—namely, the empty string—whereas $\emptyset$ represents the language that doesn't contain any strings.



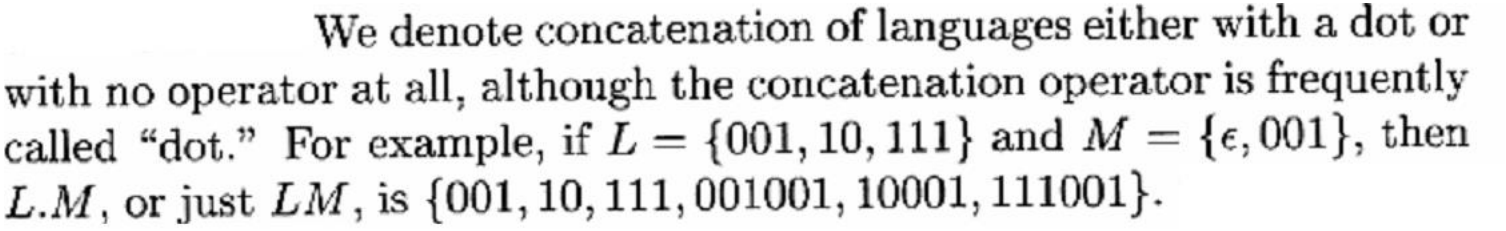
Parentheses in an expression may be omitted. If they are, evaluation is done in the precedence order: star, then concatenation, then union.

Union of Regular Expression



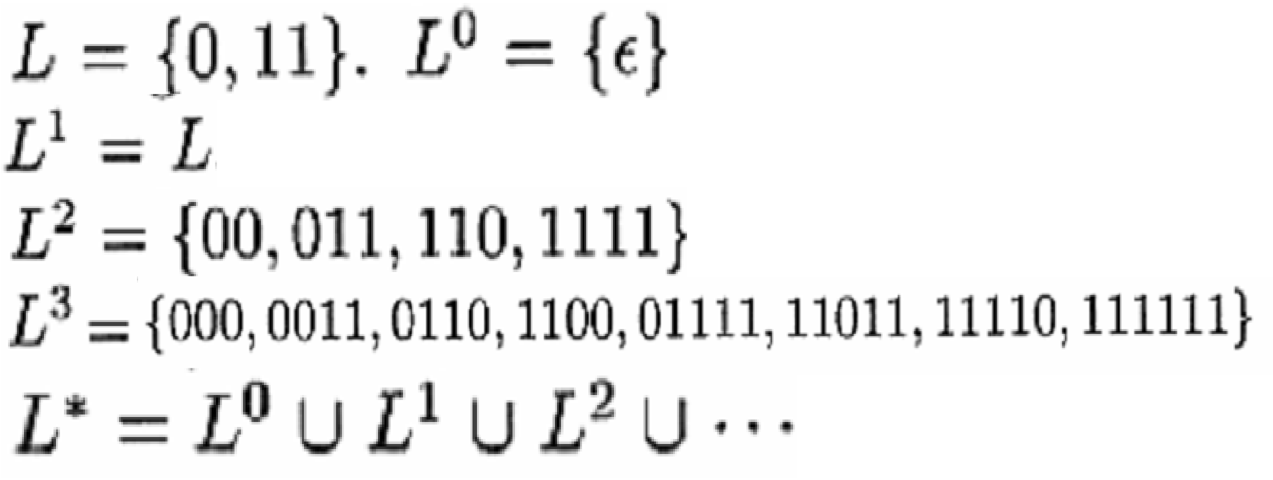
The *union* of two languages L and M, denoted $L \cup M$, is the set of strings that are in either L or M, or both. For example, if $L = \{001, 10, 111\}$ and $M = \{\epsilon, 001\}$, then $L \cup M = \{\epsilon, 10, 001, 111\}$.

Concatenation of Regular Expression



We denote concatenation of languages either with a dot or with no operator at all, although the concatenation operator is frequently called "dot." For example, if $L = \{001, 10, 111\}$ and $M = \{\epsilon, 001\}$, then $L.M$, or just LM, is $\{001, 10, 111, 001001, 10001, 111001\}$.

Closure (*) of Regular Expression



$L = \{0, 11\}$. $L^0 = \{\epsilon\}$

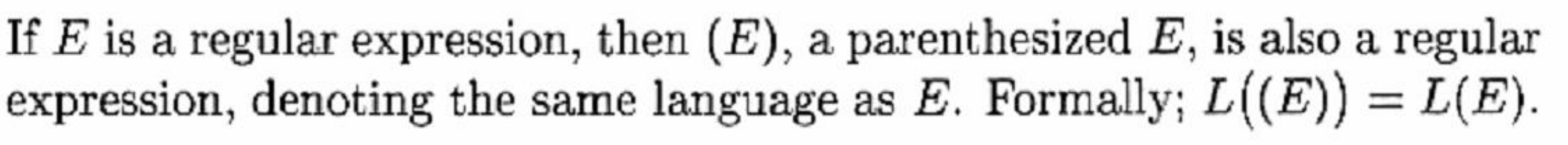
$L^1 = L$

$L^2 = \{00, 011, 110, 1111\}$

$L^3 = \{000, 0011, 0110, 1100, 01111, 11011, 11110, 111111\}$

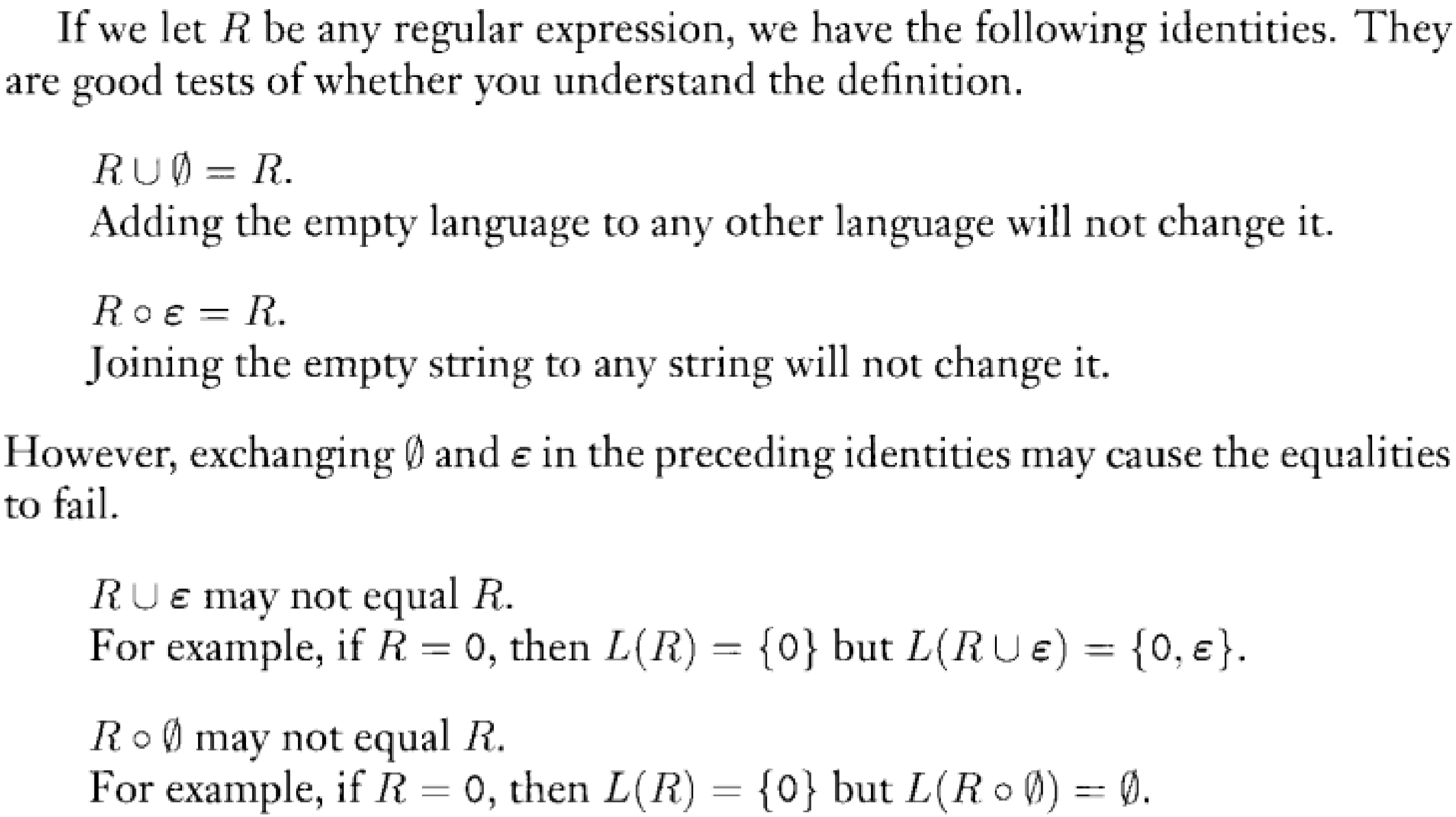
$L^* = L^0 \cup L^1 \cup L^2 \cup \cdots$

Parenthesis of Regular Expression



If E is a regular expression, then (E), a parenthesized E, is also a regular expression, denoting the same language as E. Formally; $L\big((E)\big) = L(E)$.

Note



If we let R be any regular expression, we have the following identities. They are good tests of whether you understand the definition.

$R \cup \emptyset = R$.
Adding the empty language to any other language will not change it.

$R \circ \varepsilon = R$.
Joining the empty string to any string will not change it.

However, exchanging $\emptyset$ and ε in the preceding identities may cause the equalities to fail.

$R \cup \varepsilon$ may not equal R.
For example, if $R = \mathtt{0}$, then $L(R) = \{\mathtt{0}\}$ but $L(R \cup \varepsilon) = \{\mathtt{0}, \varepsilon\}$.

$R \circ \emptyset$ may not equal R.
For example, if $R = \mathtt{0}$, then $L(R) = \{\mathtt{0}\}$ but $L(R \circ \emptyset) = \emptyset$.

R+ Notation

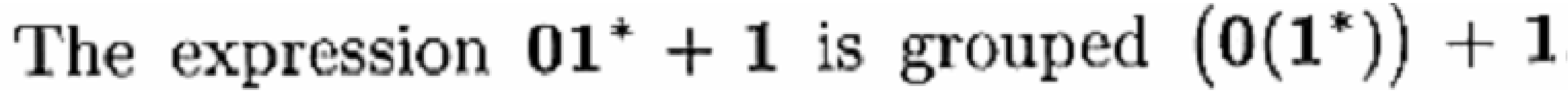
We will use r^+ as a shorthand for rr^*, i.e., denoting concatination of 1 or more strings from r.

Q: What is $r^+ \cup \varepsilon$?

Similarly, we will use r^k as a shorthand for $\underbrace{rr\ldots r}_{k}$.

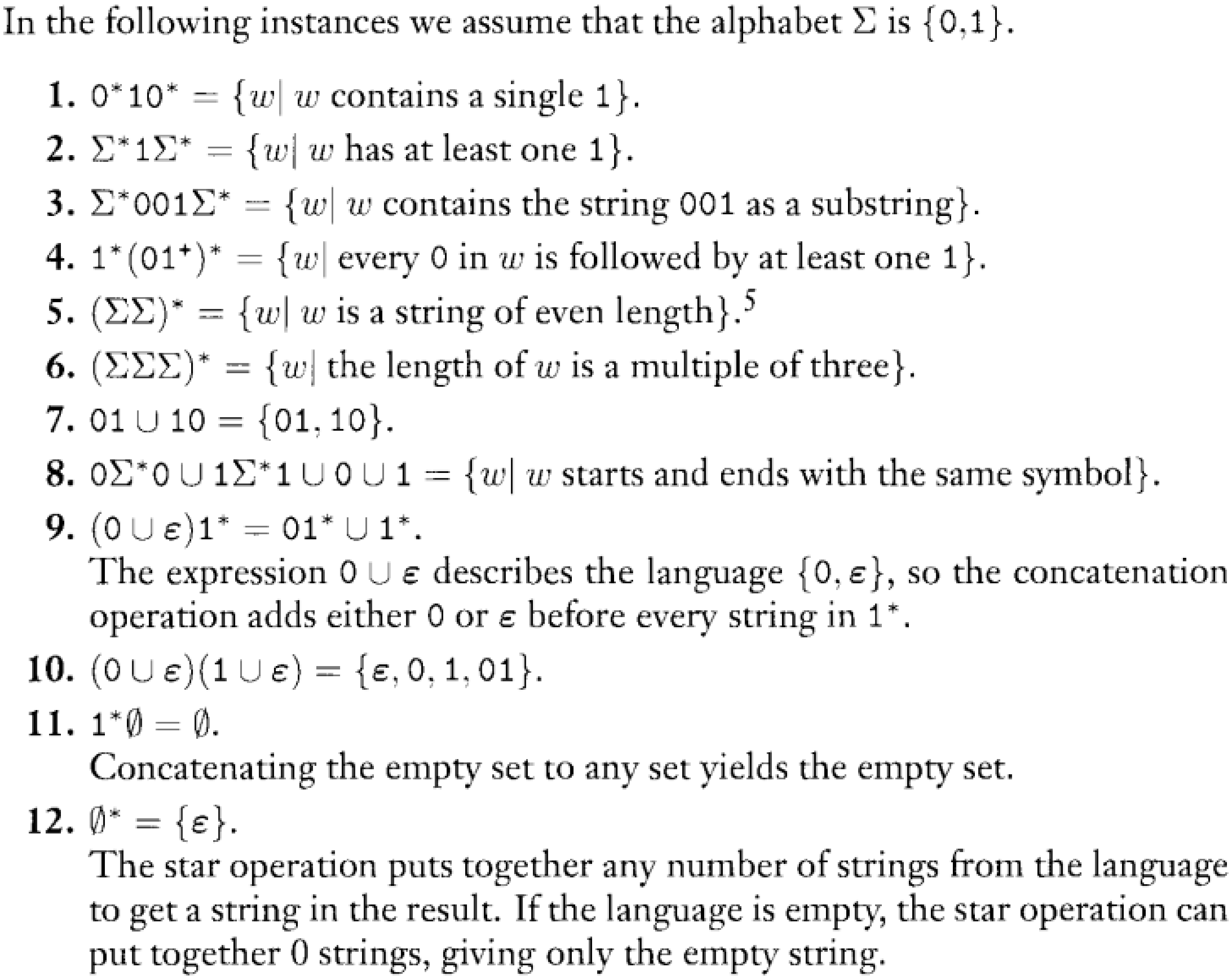
Precedence of Operation

- Parentheses may be used wherever needed to influence the grouping of operators.
- Order of precedence is * (highest),
- Then concatenation (dot .)
- Then + (lowest).



The expression $\mathbf{01}^* + \mathbf{1}$ is grouped $(\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}$

Examples



In the following instances we assume that the alphabet Σ is $\{0,1\}$.

1. $0^*10^* = \{w \mid w \text{ contains a single } 1\}$.
2. $\Sigma^*1\Sigma^* = \{w \mid w \text{ has at least one } 1\}$.
3. $\Sigma^*001\Sigma^* = \{w \mid w \text{ contains the string } 001 \text{ as a substring}\}$.
4. $1^*(01^+)^* = \{w \mid \text{every } 0 \text{ in } w \text{ is followed by at least one } 1\}$.
5. $(\Sigma\Sigma)^* = \{w \mid w \text{ is a string of even length}\}$.[5]
6. $(\Sigma\Sigma\Sigma)^* = \{w \mid \text{the length of } w \text{ is a multiple of three}\}$.
7. $01 \cup 10 = \{01, 10\}$.
8. $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1 = \{w \mid w \text{ starts and ends with the same symbol}\}$.
9. $(0 \cup \varepsilon)1^* = 01^* \cup 1^*$.
   The expression $0 \cup \varepsilon$ describes the language $\{0, \varepsilon\}$, so the concatenation operation adds either 0 or ε before every string in 1^*.
10. $(0 \cup \varepsilon)(1 \cup \varepsilon) = \{\varepsilon, 0, 1, 01\}$.
11. $1^*\emptyset = \emptyset$.
    Concatenating the empty set to any set yields the empty set.
12. $\emptyset^* = \{\varepsilon\}$.
    The star operation puts together any number of strings from the language to get a string in the result. If the language is empty, the star operation can put together 0 strings, giving only the empty string.

Example

The set of strings over {0,1} that end in 3 consecutive 1's.

$$(0 \mid 1)^* 111$$

OR

$$(0 + 1)^* 111$$

The set of strings over {0,1} that have at least one 1.

?

Example

The set of strings over {0,1} that end in 3 consecutive 1's.

$(0 \mid 1)^* 111$

OR

$(0 + 1)^* 111$

The set of strings over {0,1} that have at least one 1.

$0^* 1 (0 + 1)^*$

Example

The set of strings over {0,1} that have at most one 1.

0* | 0* 1 0*

The set of strings over {A..Z,a..z} that contain the word "main".

Let <letter> = A | B | ... | Z | a | b | ... | z

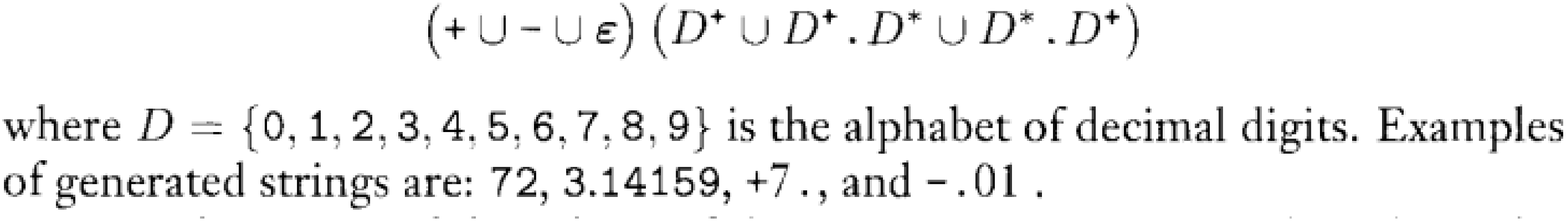
<letter>* main <letter>*

Example

The set of strings over {A..Z,a..z} that contain 3 y's.

<letter>* y <letter>* y <letter>* y <letter>*

Example



$$(+ \cup - \cup \varepsilon)\,(D^+ \cup D^+ . D^* \cup D^* . D^+)$$

where $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ is the alphabet of decimal digits. Examples of generated strings are: 72, 3.14159, +7., and -.01 .

Example

Consider ∑ = { a }

L is a language that each word is of odd length

a(aa)*

Example

Let $\Sigma = \{0, 1\}$.

$0^*10^* = \{w \mid w \text{ contains a single } 1\}$
$\Sigma^*1\Sigma^* = \{w \mid w \text{ contains at least one } 1\}$

$(\Sigma\Sigma)^* = \{w \mid \text{the length of } w \text{ is even }\}$
$(\Sigma\Sigma\Sigma)^* = \{w \mid \text{the length of } w \text{ is multiple of } 3\}$

Q: What is $(01^+)^*$?

$0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1 = \{w \mid$
w starts and ends with the same symbol $\}$
$(0 \cup \varepsilon)1^* = 01^* \cup 1^*$
$(0 \cup \varepsilon)(1 \cup \varepsilon) = \{\varepsilon, 0, 1, 01\}$

Example

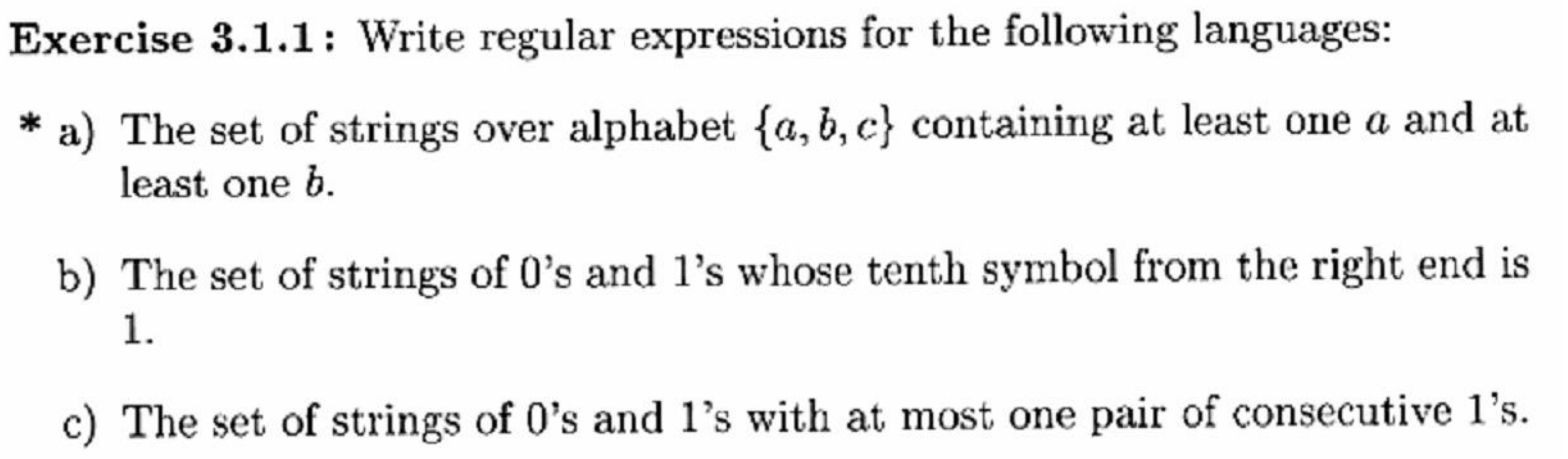
- L(**01**) = {01}.
- L(**01**+**0**) = {01, 0}.
- L(**0**(**1**+**0**)) = {01, 00}.
  - Note order of precedence of operators.
- L(**0***) = {ε, 0, 00, 000,… }.
- L((**0**+**10**)*(ε+**1**)) = all strings of 0's and 1's without two consecutive 1's.

Task

Construct a RE over Σ={0,1} such that

1) It contains all strings that have two consecutive “0”s.
2) It contains all strings except those with two consecutive “0”s.
3) It contains all strings with an even number of “0”s.

Task



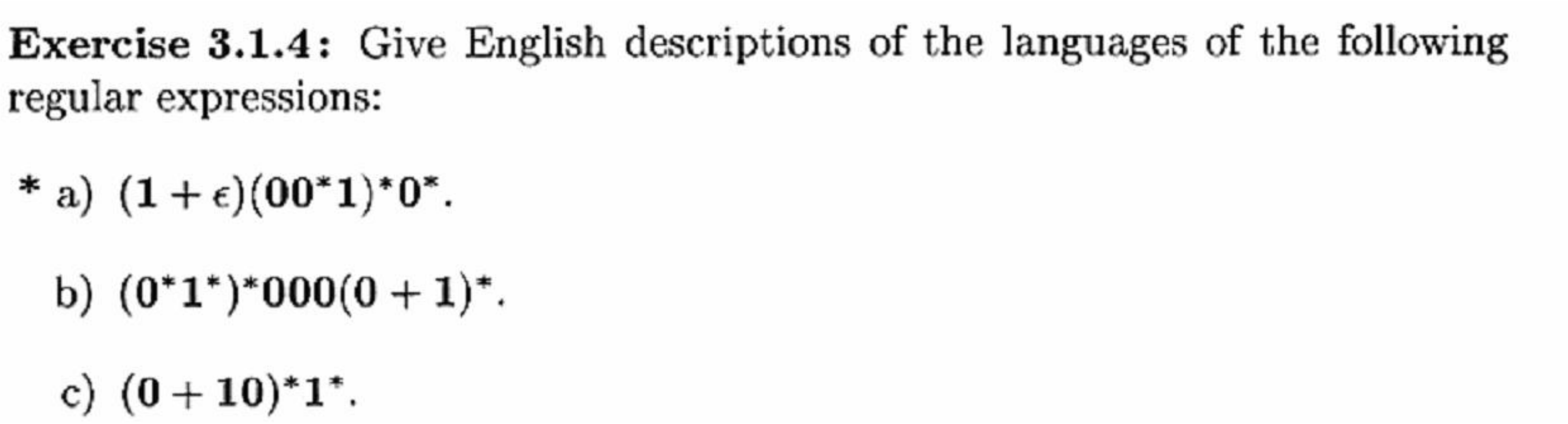
Exercise 3.1.1: Write regular expressions for the following languages:

* a) The set of strings over alphabet $\{a, b, c\}$ containing at least one a and at least one b.

b) The set of strings of 0's and 1's whose tenth symbol from the right end is 1.

c) The set of strings of 0's and 1's with at most one pair of consecutive 1's.

Task

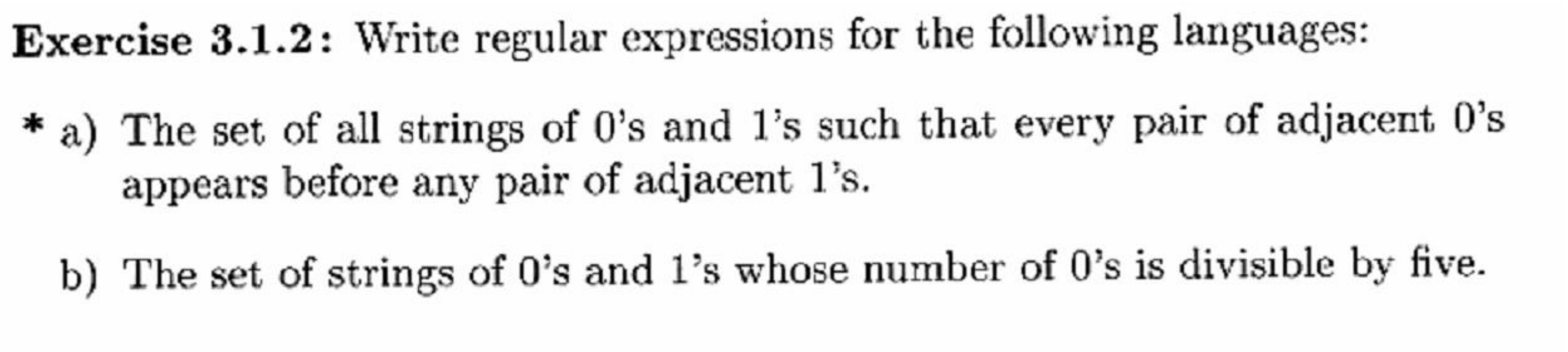


Exercise 3.1.4: Give English descriptions of the languages of the following regular expressions:

* a) $(\mathbf{1}+\epsilon)(\mathbf{00}^*\mathbf{1})^*\mathbf{0}^*$.

b) $(\mathbf{0}^*\mathbf{1}^*)^*\mathbf{000}(\mathbf{0}+\mathbf{1})^*$.

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



Exercise 3.1.2: Write regular expressions for the following languages:

* a) The set of all strings of 0's and 1's such that 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 five.