1. Computation Theory:

Theory of Computation, also known as the Automata theory, is a field within computer science and mathematics that focuses on studying abstract machines to understand the capabilities and limitations of computation by analyzing mathematical models of how machines can perform calculations and solve the problems. They can be simple as estimation for driving time between cities, and as complex as a weather prediction.

2. Alphabet, String, and Language:

The ability to represent information is crucial to communicating and processing information. The English language, for example, in its spoken form relies on some finite set of basic sounds as a set of primitives. The words are defined in term of finite sequences of such sounds. Sentences are derived from finite sequences of words. Paragraphs are obtained from finite sequences of sentences.

Similar approaches have been developed also for representing elements of other sets. For instance, the natural number can be represented by finite sequences of decimal digits (0,1,2,3,4,5,6,7,8,9).

Computations, like natural languages, are expected to deal with information in its most general form. Consequently, computations function as manipulators of integers, graphs, programs, and many other kinds of entities. However, in reality computations only manipulate strings of symbols that represent the objects.

An alphabet: is a finite, nonempty set of symbols. Its symbol is Σ(sigma) for an alphabet. Common alphabet includes:

1. Σ={0,1}, the binary alphabet.
2. Σ={a,b,….,z}, the set of all lower-case letters.
3. The set of all ASCII character, or the set of all printable ASCII characters.

Strings: is a finite sequence of symbols chosen from some alphabet. For example: 01101 is a string from the binary alphabet Σ={0,1}. The string 111 is another string chosen from this alphabet.

1. The empty string: is the string with zero occurrences of symbols. This string denoted ϵ, is a string that may be chosen from any alphabet.

2. Length of a string: is the number of positions for symbols in the string.

For instance 01101 has length 5. It's common to say that the length of a string is "the number of symbols"in the string; this statement is accepted but not correct. Thus, there are only two symbols, 0 and 1, in the string 01101, but there are five positions for symbols, and its length is 5.

The standard notation for the length of a string w is $\|w\|$.

for example, $|011|=3$ and $|\epsilon|=0$.

3. Concatenation of strings

Let x and y be strings. Then xy denotes the concatenation of x and y, that is, the string formed by making a copy of x and following it by a copy of y.

Example 1:

let x =01101 and y=110. Then xy=01101110 and yx=11001101. For any string w, the equations $\epsilon w=w\epsilon=w$ hold. That is, ϵ is the identity for concatenation, since when cocncatenated with any string it yields the other string as a result.

Formal Languages:

A set of strings all of which are chosen from some $\sum^*$, where $\sum$ is a particular alphabet, is called a language.

If $\sum$ is an alphabet, and L is a subset of $\sum^*$, then L is a language over $\sum$. Notice that a language over $\sum$ need not include strings with all symbols of $\sum$, so once established that L is a language over $\sum$, it is a language over any alphabet that is a superset of $\sum$ is know.

The choice of the term "language " may seem strange. However, common languages can be viewed as sets of strings. An example is English, where the collection of legal English words is a set of strings over the alphabet that consists of all the letters.

Another example is C, or any other programming language, where the legal programs are a subset of the possible strings that can be formed from the alphabet of the language. This alphabet is a subset of the ASCII characters. The exact alphabet may differ slightly among different programming language, but generally includes the upper – and lower-case letters, the digits, punctuation, and mathematical symbols.

Some are abstract example, such as:

1. The languages of all strings consisting of n 0's followed by n 1's, for some $n \geq 0$: {ϵ,01, 0011, 000111, …}.
2. The set of strings of 0's and 1's with an equal number of each: {ϵ, 01, 10, 0011, 0101, 1001,1100,1010,0110,…}.
3. The set of binary numbers whose value is a prime: {10, 11, 101, 111, 1011, …}.

4. $\sum^*$ is a language for any alphabet $\sum$.
5. Φ, the empty language, is a language haven't any alphabet
6. $\{\epsilon\}$, the language consisting of only the empty string, is also a language over any alphabet. Notice that $\varphi \neq \{\epsilon\}$.; the former has no strings and the latter has one string.

NOTE: The only important constraint on what can be a language is that all alphabets are finite.

Since languages are sets, it can be combined by the set operations of :

1. Union, The union of two languages L1 and L2, denoted L1∪ L2, refers to the language that consists of all the strings that are either in L1 or in L2, that is, to { x | x is in L1 or x is in L2 }.

2. Intersection, The intersection of L1 and L2, denoted L1∩ L2, refers to the language that consists of all the strings that are both in L1 and L2, that is, to
{ x | x is in L1 and in L2 }.

3. *Difference*. The difference of L_1 and L_2, denoted L_1 - L_2, refers to the language that consists of all the strings that are in L_1 but not in L_2, that is, to
{ x | x is in L_1 but not in L_2 }.

4. The *complementation* of a language L over $\sum$, denoted $\bar{L}$, refers to the language that consists of all the strings over $\sum$ that are not in L, that is, to
{ x | x is in $\sum^*$ but not in L }.

Example 2: Consider the languages $L_1 = \{\epsilon, 0, 1\}$ and $L_2 = \{\epsilon, 01, 11\}$. The union of these languages is $L_1 \cup L_2 = \{\epsilon, 0, 1, 01, 11\}$, their intersection is
$L_1 \cap L_2 = \{\epsilon\}$, and the complementation of $\overline{L1} = \{00, 01, 10, 11, 000, 001, \ldots\}$.
L1-L2={0,1}, L2-L1={01,11}
L1.L2={ ϵ, 01, 11, 0, 001,011, 1, 101, 111}
L1*={ ϵ, 0,1, 01, 10,00, 11,…}

$\emptyset \cup L = L$ for each language L. Similarly, $\emptyset \cap L = \emptyset$ for each language L. On the other hand, $\bar{\emptyset} = \sum^*$ and $\overline{\sum^*} = \emptyset$ for each alphabet.

In addition, certain operations are meaningful only for languages. The first of these is the *concatentation* of languages. If L_1 and L_2 are languages over $\sum$, their concatenation is $L=L_1 o L_2$, or simply $L=L_1L_2$, where

$L=\{w \in \sum^* : w=xoy \text{ for some } x \in L_1 \text{ and } y \in L_2\}$.

Another language operation is the *Kleene star* of a language L, denoted by L^*. L^* is the set of all strings obtained by concatenating zero or more strings from L. Thus, $L^*=\{w \in \sum^* : w=w_1 o \ldots o w_k \text{ for some } k \geq 0 \text{ and some } w_1, \ldots, w_1 \in L\}$

Example: If ∑={a,b} is a language, then ∑*=∑0 U ∑1 U ∑2 U…..∑∞

= { ϵ, a, b, aa, ab, bb, ba, aaa, bbb, baa, abb, abab, ……}

∑+=∑1 U ∑2 U…..∑∞

= { a, b, aa, bb, ab, ba, aaa, bbb, abb, aba, baba, ……}

Note: ∑*-∑+={ ϵ}

∑+ - ∑* = Φ

Example: If ∑={a,b} is a language,

1. L={strings with even length over alphabet}

   Sol:

   L={ ϵ, aa, bb, ab, ba, aabb, bbaa, abab, baba, baaa,……}
2. L={strings starts with a and end with b and length <=4}

   Sol:

   L={ab, abb, aab, aabb, aaab, abbb, abab}

Example 3: if L={01, 1, 100}, then 110001110011∈L*, since
1100011100111=1o100 o01 o1 o100 o1 o1, and each of these strings is in L.

1100011100110=1o100 o01 o1 o100 o1 o1 o0 then 1100011100110∉L*,

---

Exercise 1:
Find a possible alphabet Σ for the following languages. A word foobar should be interpreted as a string of characters f, o, o, b , a and r.
(i) The language L = {oh, ouch, ugh}
(ii) The language L = {apple, pear, 4711}
(iii) The language of all binary strings

Exercise 2:
Describe what the Kleene star operation * over the following alphabets produces.
(i) Σ = {0, 1}
(ii) Σ = {a}
(iii) Σ = ∅ (the empty alphabet)

Example: L1={11,a1}, L2={0b,a}

1) L1.L2={110b, 11a, a10b, a1a}
2) L2.L1={0b11, 0ba1, a11, aa1}
3) L2.L2= {0b0b, 0ba, a0b, aa}
4) {aa}.L1={aa11, aaa1}
5) L1.L1=?
6) What strings come from L1.L2 with length =4?
   ={110b, a10b}
7) What strings come from L1.L2 with length <= 4?
8) What strings comes from L1.L1 with length >4? Φ
9) L1*= $L1^0$ U $L1^1$ U $L1^2$ U…..$L1^\infty$ ={ ϵ,11, a1, 1111 , 11a1, a1a1, a111, …..}
10) Give strings belongs to L1* with length <=3 ? ={ ϵ, 11, a1}
11) Give strings belongs to L2* with length =3?

Exercise 3:

Describe what the +-operation over the following alphabets produces.

(i) $\Sigma = \{0, 1\}$

(ii) $\Sigma = \{a\}$

(iii) $\Sigma = \varnothing$ (the empty alphabet)

Homework 1:

State the alphabet Σ for the following languages :

(i) $L = \Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, \ldots\}$

(ii) $L = \Sigma^+ = \{a, aa, aaa, \ldots\}$

(iii) $L = \Sigma^+ = \{\varepsilon\}$

Homework 2:

Assuming that $\Sigma = \{0, 1\}$, construct complement languages for the following.

(i) $\{010, 101, 11\}$

(ii) $\Sigma^* - \{110\}$

(iii) $\Sigma^+ - \varepsilon$