Lesson 3 - Alphabets, Strings and Languages

Alphabets

`Alphabet` is a non-empty finite set of symbols.

For example:

$$\Sigma_1 = \{0, 1\}$$

That is an example of alphabet with two symbols (an alphabet of `binary numbers`).

2nd example:

$$\Sigma_2 = \{a, b, c, ..., x, y, z\}$$

That is an example of alphabet with 26 symbols (an alphabet of `English alphabet`).

3rd example:

$$\Sigma_3 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

That is an example of alphabet with ten symbols (an alphabet of `decimal numbers`).

Strings

`String` is a finite sequence of symbols from an alphabet.

Let an alphabet:

$$\Sigma_1 = \{0, 1\}$$

Therefore, the strings *0*, *1* and *001* are examples of string over the alphabet above. We can also write strings as `w`. For example:

$$w = 0$$

$$w = 1$$

$$w = 001$$

Another example, let an alphabet:

$$\Sigma_2 = \{a, b\}$$

Thus, the following are strings from the second alphabet above:

$$w = a$$

$$w = ab$$

$$w = baa$$

Length of string or cardinality

`Length of string or cardinality` is the number of symbols in a string.

For example:

$$w = abcd$$

The length of the string `w` is 4, since it has 4 symbols: *a, b, c, d*. We can also denote the length of string using character `|w|`:

$$|w| = 4$$

Empty string

`Empty string` is a string that contains no symbols. In this course, we denote it with λ (lambda) symbol. Of course, `empty string` has a length of 0:

$$|\lambda| = 0$$

Concatenation

`Concatentation` combines two strings. If we have string `x` and `y` then, their concatenation is denoted by `xy`, it is the string formed by appending string `y` to `x`. For example, say we have two strings:

$$x = louis$$

$$y = facun$$

Therefore `xy` is:

$$xy = louisfacun$$

Reversely, `yx` is:

$$yx = facunlouis$$

If we concatenate an empty string some string, then that string won't change. For example:

$$x\lambda = x$$

Substring

A string `x` is a `substring` of some string `y` if the string `x` appears `consecutively` in string `y`.

For example:

$$y = banana$$

$$x = nana$$

The string `nana` appears consecutively in the string `banana`. Thus, the string `x` is a substring of string `y`. But the string `nab` is *not* a substring of `y`.

Suffix and prefix

A string x is a `suffix` of y if it is a substring of y while also appearing in the end of y. For example:

$$y = banana$$

$$x = nana$$

The x is a substring of y and it also appears at the end of y. Thus, x is a `prefix` of y. Similarly, it is a substring but it appears at the `start`, then it is called a `suffix`. For example:

$$y = banana$$

$$z = bana$$

In this case, the string z or bana is a substring of y and also appears at the start. Therefore z is a `prefix` of y.

k copies of w

$$w^k$$

Represents the concatenation of string w, k times. For example:.

$$w = ab$$

If k is equal to 1, then the string w is concatenated 1 time:

$$w^1 = ab$$

If k is equal to 2, then:

$$w^2 = abab$$

And so on:

$$w^3 = ababab$$

$$w^4 = abababab$$

Languages

Language `(L)` is a set of all possible strings from an alphabet, given a condition.

For example, given an alphabet:

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

With a condition: `No two consecutive 1s`, then, the language is:

$$L = \{\lambda, 0, 1, 00, 01, 10, 000, 001, 010, 100, ...\}$$

Note that, we also included `empty string` since this is also a `possible string`. Also, in the set, we won't put strings such as: `11`, `011`, `110`, `111` and so on, since all these strings contains two

consecutives `1s`.

Language notation

Most of the time it would be use to use `set notation` to define a language instead of tediously writing it.

$$w = ab$$

Say, you have language:

$$L = \{ab, abab, ababab, abababab, ...\}$$

We can simply say using notation:

$$L = \{w^k | k \geq 1\}$$

L is k copies of w such that k is greater than or equal to 1.

Summary

Symbol/Notation	Description
$\{...\}$	Sets
Σ	Alphabet
w	String
λ	Empty string
$\|w\|$	cardinality of string `w`
xy	x string concatenated to y
w^k	k copies of `w`