



**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE CÓMPUTO**



**U.A: Teoría Computacional**

**“Reporte práctica 01”**

**Grupo: 2CV13**

**Profesor: De la O Torres Saul**

**Alumno: Sánchez Becerra Ernesto Daniel**

## **Introducción:**

### **¿Qué es un alfabeto?**

Un alfabeto es un conjunto finito no vacío cuyos elementos se llaman símbolos. Denotamos un alfabeto arbitrario con la letra  $\Sigma$ . Una cadena o palabra sobre un alfabeto  $\Sigma$  es cualquier sucesión finita de elementos de  $\Sigma$ .

### **¿Qué es un lenguaje?**

Conjunto de cadenas elegidas de algún alfabeto.

– Nota: El lenguaje puede ser infinito, pero existe algún conjunto finito de símbolos de los cuales se componen todas sus cadenas.

– El conjunto de todas las cadenas binarias que consisten en algún número de 0's seguidos por un número igual de 1's; esto es:  $\{0, 01, 0011, 000111, \dots\}$ .

– C (el conjunto de programas compilables en C).

Una manera importante de describir ciertos lenguajes simples, pero altamente útiles es la llamada "lenguajes regulares".

- Un grafo con un número finito de nodos, llamados estados.
- Los arcos se etiquetan con uno o más símbolos de algún alfabeto.
- Un estado es designado como el estado de comienzo ó estado inicial.
- Algunos estados son estados finales o estados de aceptación.
- El lenguaje de los AF (Autómata Finito) es el conjunto de cadenas que etiquetan rutas que van desde el estado inicial a algún estado de aceptación.

### **¿Qué es una cadena?**

secuencia finita de símbolos elegidos de un alfabeto.

– 01101

– abracadabra

– La cadena vacía se denota como:  $\epsilon$ ,

– Todas las cadenas de un alfabeto  $\Sigma$  de longitud  $k$  se denotan como  $\Sigma^k$ . E.g.,  $\Sigma^0 = \{\epsilon\}$ , si  $\Sigma = \{0, 1\}$ , entonces,  $\Sigma^1 = \{0, 1\}$ ,  $\Sigma^2 = \{00, 01, 10, 11\}$ , etc.

– El conjunto de todas las cadenas de un alfabeto  $\Sigma$  se denota como  $\Sigma^*$ .  $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$ . Sin la cadena vacía:  $\Sigma^+$ . Por lo que:  $\Sigma^* = \Sigma^0 \cup \Sigma^+$ .

### **¿Qué es un prefijo?**

Los prefijos están formados por los primeros símbolos de la cadena; y los sufijos, por los últimos. Un prefijo o sufijo de una cadena que no sea la misma cadena es un prefijo o sufijo propios

### **¿Qué es un sufijo?**

Es un arreglo ordenado de todos los sufijos de una cadena dada. Esta estructura de datos es muy simple, sin embargo, es muy poderosa y es usada en algoritmos de compresión de datos y dentro del campo de la bioinformática, indización de textos completos, entre otros.

**Objetivo:**

Realizar un programa que calcule TODOS los prefijos, sufijos y subcadenas de una cadena determinada. La longitud máxima de la cadena de entrada deberá ser con un máximo de 100 caracteres, el alfabeto de 30 caracteres y ambos se piden como entrada del programa al usuario.

**Desarrollo Experimental:**

Para el desarrollo de esta práctica se utilizó el entorno de programación "NetBeans" pues con él se puede trabajar cómodamente con archivos de tipo Java.

Empezamos creando un nuevo proyecto llamado "Programa01". En él creamos 2 clases, la primera es nuestra clase principal que se genera al momento de realizar un nuevo proyecto y la segunda clase llamada "Cadena" que contiene los métodos que se ocuparon durante el desarrollo de la práctica y que a continuación, presentaremos y explicaremos.

**1- Public Cadena:**

Este método es el constructor de nuestra clase "Cadena", básicamente sirve para validar un dato de tipo String como un dato de tipo cadena para poder trabajar con esa cadena y hacer las operaciones que se realizan en la práctica. Esta clase está contenida en el archivo "Cadena.java"

**2- Public boolean equals:**

Este método es el que se encarga de hacer la comparación entre cadenas, cuando definimos la cadena que conforma nuestro alfabeto y queremos corroborar que la subCadena que ingresamos es válida, se utiliza este método y facilita esta comparación entre cadenas. Este método se encuentra contenido en el archivo "Cadena.java".

**3- Public String toString:**

Este método se encarga de devolver una cadena, ya sea la cadena que estamos solicitando como son las subCadenas, prefijos y sufijos o bien, devuelve el conjunto vacío si no se agregó alguna cadena para ser analizada después de que se estableció el alfabeto. Este método se encuentra contenido en el archivo "Cadena.java".

**4- Public int length:**

Este método se encarga de obtener la longitud de las cadenas que estemos analizando, en este caso, se emplea cuando queremos determinar los prefijos, sufijos y subCadenas. Este método se encuentra contenido en el archivo "Cadena.java".

**5- Public Cadena [] obtenSubcadena:**

Este método se encarga de leer la subCadena que se pide al usuario y a partir de ella, calcula las demás subcadenas que están contenidas en la cadena que el usuario ingresa y que ya fue validada después de definir el alfabeto. Este método se encuentra contenido en el archivo "Cadena.java".

**6- Public Cadena [] obtenerPrefijo:**

El siguiente método se encarga de calcular los prefijos, esto se lleva a cabo de acuerdo a la palabra que se ingresa por el usuario para calcular los prefijos a partir de ella, es importante ingresar una cadena válida pues en caso de no hacerlo, el método no realiza adecuadamente el cálculo. Este método se encuentra contenido en el archivo "Cadena.java".

**7- Public Cadena [] obtenerSufijo:**

Finalmente, este método se encarga de calcular los sufijos, esto se lleva a cabo en función de la palabra que el usuario ingresa desde teclado. Al igual que en el método

de los prefijos, se debe de validar primero la cadena que ingresamos para calcular los sufijos, en caso contrario, no se podrá cumplir el cálculo de los sufijos de manera adecuada. Este método se encuentra contenido en el archivo "Cadena.java".

**Nota:** El proyecto creado en NetBeans se encuentra anexo en la carpeta donde está contenido este reporte.

### **Conclusión:**

Con la elaboración de esta práctica se pusieron a prueba los temas vistos en clase acerca de los lenguajes y operaciones entre cadenas que se pueden realizar a partir de un alfabeto dado.

Puedo concluir que es importante entender la teoría antes de iniciar con la práctica pues si no se entienden los conceptos y en este caso particular, las definiciones formales y experimentales de los temas, es un poco complicado pensar en cómo se pueden programar las clases que resuelvan el objetivo de la práctica.