



**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO**



**UNIDAD DE APRENDIZAJE**

**TEORÍA COMPUTACIONAL**

**TAREA**

**PRÁCTICA 2**

**ALUMNO**

**BARRERA ESTRELLA EMANUEL**

**PROFESOR**

**JUÁREZ MARTÍNEZ GENARO**

**GRUPO**

**2CM1**

## Introducción

Un lenguaje es un conjunto de todas las cadenas de las elegidas en  $\Sigma^*$ , donde  $\Sigma$  es un alfabeto en particular, llamado *lenguaje*. Si  $\Sigma$  es un alfabeto, y  $L \subseteq \Sigma^*$ , entonces  $L$  es un lenguaje sobre  $\Sigma$ . Nótese que un lenguaje sobre  $\Sigma$  no necesariamente incluye todas las cadenas con todos los símbolos de  $\Sigma$ , entonces una vez que hemos establecido que  $L$  es un lenguaje sobre  $\Sigma$ , también sabremos si es un lenguaje sobre cualquier alfabeto que es superconjunto de  $\Sigma$ .

Como sea el caso, existen muchos otros lenguajes que aparecen cuando estudiamos a los autómatas. Algunos son ejemplos abstractos, como son:

1. El lenguaje de todas las cadenas que consisten de  $n$  0's seguidos por  $n$  1's, para cualquier  $n \geq 0$  :  $\{\varepsilon, 01, 0011, 000111, \dots\}$ .
2. El conjunto de cadenas de 0's y 1's con un número igual de cada uno:

$$\{\varepsilon, 01, 10, 0011, 0101, 1001, \dots\}$$

3. El conjunto de las cadenas binarios cuyo valor es un número primo:

$$\{\varepsilon, 10, 11, 101, 111, 1011, \dots\}$$

4.  $\Sigma^*$  es un lenguaje para cualquier alfabeto  $\Sigma$ .
5.  $\emptyset$ , es lenguaje vacío para cualquier alfabeto  $\Sigma$ .
6.  $\{\varepsilon\}$ , el lenguaje que consiste por sólo la cadena vacía, es también un lenguaje sobre cualquier alfabeto. Nótese que  $\emptyset \neq \{\varepsilon\}$ ; una no tiene cadenas y la otra tiene una cadena.

La única restricción importante en qué puede ser un lenguaje es que todos los alfabetos son finitos. Así, los lenguajes aunque tienen un número infinito de cadenas, están restringidos de consistir sólo por cadenas en base a un alfabeto finito.

## Desarrollo

La práctica que se presenta a continuación es de un programa que pide un número entero  $n$  tal que,  $0 \leq n \leq 100,000$ , para calcular el lenguaje del conjunto de las cadenas binarios cuyo valor es un número primo hasta  $n$ .

Implementando este programa en lenguaje C, se tiene el siguiente código:

```
int main ()
{
    long num, total, primo, limite, i, *listaprimos;
    int es_primo, opc;
    long o;
    char *cadena, *s_opc, *s_n;
    FILE *archivo;
    listaprimos = (long *) calloc (1, sizeof (long));
    listaprimos[0] = 2;
    do
    {
        printf ("Generador del subconjunto L de Sigma, ¿si_w_en_L es un_numero_primo\n");
        printf ("1) Generar el subconjunto L de manera manual\n");
        printf ("2) Generar el subconjunto L de manera aleatoria\n");
        printf ("3) Salir\n");
        scan (&s_opc);
    }
    while (!(cadenaEsNumero (&s_opc)));
    opc = cadenaANumero (&s_opc);
```

Como se puede ver, este programa tiene un menú que nos permite elegir cómo se genera el lenguaje  $L$ , si de forma manual o de forma automática.

```
        printf ("Ingrese el numero primo hasta cual generar el subconjunto L:\n");
    do
    {
        scan (&s_n);
    }
    while (!(cadenaEsNumero (&s_n)));
    num = cadenaANumeroL (&s_n);
```

De esta forma estamos generando el número de forma manual.

```
    srand (time (NULL));  
    //numero aleatorio entre 1 y 100,000  
    num =(long) (rand () % (100001));  
    printf ("Subconjunto_L_generado_aleatoriamente_hasta_el_numero_%ld\n",num);
```

De esta forma generamos el número de forma automática con un random().

```
system ("rm_primos.txt");  
system ("echo >_primos.txt");  
archivo = fopen ("primos.txt", "w");
```

Creamos el fichero donde se escribe la salida del programa.

El algoritmo para calcular números primos dado un límite, donde va añadiendo los números primos a una lista, verifica si los números que siguen, son compuestos a partir de los números primos enlistados en las posiciones anteriores a este, si es un número primo, el programa añade este número a la lista y así sucesivamente.

```

i = 3;
total = 0;
if (i >= 2)
{
    fprintf (archivo , "{");
    numeroABinario (&cadena, 2);
    fprintf (archivo , "%s", cadena);
    while (i <= num)
    {
        limite = ((long) sqrt (num)) + 1;
        es_primo = 1;
        for (primo = 0; primo <= total; primo++)
        {
            if (i % listaprimos[primo] == 0)
            {
                es_primo = 0;
                break;
            }
            if (listaprimos[primo] > limite)
            {
                break;
            }
        }
        if (es_primo)
        {
            total++;
            listaprimos =
                (long *) realloc (listaprimos , sizeof (long) * (total + 1));
            listaprimos[total] = i;
            fprintf (archivo , ",_");
            numeroABinario (&cadena, i);
            fprintf (archivo , "%s", cadena);
        }
        i += 2;
    }
}

```

```

void numeroABinario (char **cadena, long n)
{
    int i, c;
    char *rev;
    i = 0;
    *cadena = (char *) malloc (sizeof (char));
    rev = (char *) malloc (sizeof (char));
    if (n == 0)
    {
        (*cadena)[i] = '0';
        i++;
        *cadena = (char *) realloc (*cadena, sizeof (char) * (i + 1));
        rev = (char *) realloc (rev, sizeof (char) * (i + 1));
    }
    while (n > 0)
    {
        (*cadena)[i] = (n % 2) + 48;
        n /= 2;
        i++;
        *cadena = (char *) realloc (*cadena, sizeof (char) * (i + 1));
        rev = (char *) realloc (rev, sizeof (char) * (i + 1));
    }
    i--;
    c = 0;
    while (i >= 0)
    {
        rev[c] = (*cadena)[i];
        c++;
        i--;
    }
    rev[c] = '\0';
    i = 0;
    while (rev[i] != '\0')
    {
        (*cadena)[i] = rev[i];
        i++;
    }
    (*cadena)[i] = '\0';
}

```

Como adicional, está este método casero para convertir un número decimal a una cadena con el valor binario de dicho número.

La ejecución sería:

```
emanuel_9809@emanuel-98:~/Dropbox/teoria/p2$ ./a.out
Generador del subconjunto L de Sigma, si w en L es un numero primo
1)Generar el subconjunto L de manera manual
2)Generar el subconjunto L de manera aleatoria
3)Salir
2
Subconjunto L generado aleatoriamente hasta el numero 23274
emanuel_9809@emanuel-98:~/Dropbox/teoria/p2$
```

```
emanuel_9809@emanuel-98:~/Dropbox/teoria/p2$ ./a.out
Generador del subconjunto L de Sigma, si w en L es un numero primo
1)Generar el subconjunto L de manera manual
2)Generar el subconjunto L de manera aleatoria
3)Salir
2
Subconjunto L generado aleatoriamente hasta el numero 23274
emanuel_9809@emanuel-98:~/Dropbox/teoria/p2$ ./a.out
Generador del subconjunto L de Sigma, si w en L es un numero primo
1)Generar el subconjunto L de manera manual
2)Generar el subconjunto L de manera aleatoria
3)Salir
1
Ingrese el numero primo hasta cual generar el subconjunto L:
100
emanuel_9809@emanuel-98:~/Dropbox/teoria/p2$
```

Después de generar los números primos en el fichero, procedemos a revisar el archivo con la salida del programa.

p2.c ×

p2.c ×

p1.c ×

primos.txt ×

```
{10, 11, 101, 111, 1011, 1101, 10001, 10011, 10111, 11101, 11111, 100101, 101001, 101011, 101111, 110101, 111011, 111101, 1000011, 1000111, 1001001, 1001111, 1010011, 1011001, 1100001}
```

Texto plano ▾ Anchura de la pestaña: 8 ▾ Ln 1, Col 1 ▾ INS

En este archivo de texto, se encuentra el lenguaje.