IMPLEMENTACIÓN DE UN ANALIZADOR LÉXICO

La implementación del AL en un caso genérico sería como se indica a continuación.

/ Se tiene la matriz de transición (MT_AFD) correspondiente al AFD, en la que se incluyen también las acciones semánticas y los errores.*
En la Inicialización del Procesador, entre otras cosas, se ha abierto el fichero de entrada y se ha leído el primer carácter de dicho fichero (car:= leer())/*

```
ALex ()
{
  estado:= 0;  /* estado inicial */
  LOOP until estado final
  {
      acción:= MT_AFD.acción (estado, car);
      estado:= MT_AFD.estado (estado, car);
      IF estado = null
        then error(acción)
        else switch acción      /* ejecuta las acciones */
        {
          case Ai: Acción semántica
          …
        }
  } /* fin del LOOP */
}
```

Veamos un ejemplo concreto. Supongamos un lenguaje con:

- Números enteros hexadecimales: 0x seguido de al menos un dígito hexadecimal (0-9, A-F) que se representan en 4 bytes
- Operadores + y –
- Los elementos pueden ir separados por espacios
- Los números enteros pueden ser positivos o negativos

El diseño incluye identificar los *tokens*, escribir la GR, obtener el AFD que completaremos con las acciones semánticas y hacer el tratamiento de los errores.

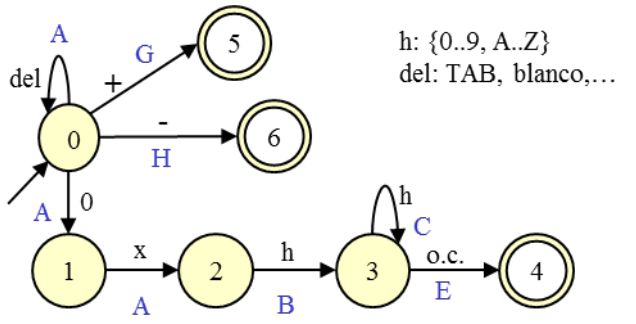
Los *tokens* son:

- Número enteros: <Entero, valor>
- Operadores aritméticos: <Más, -> y <Menos, ->.

La gramática regular que genera estos *tokens* es:

S → 0 A | + | - | del S
A → x B
B → h C
C → h C | λ

El AFD + acciones semánticas para el ejemplo puede ser:




```
A:  Lee
B:  valor:= valorHexadecimal (h); Lee
C:  valor:= valor * 16 + valorHexadecimal (h)
    Lee
E:  if valor < 2^31
        Then GenToken (Entero, valor)
        Else Error (61)
G:  Lee; GenToken (Más, -)
H:  Lee; GenToken (Menos, -)
```

Representado el AFD como una matriz de transición (incluyendo las acciones y los errores) tendríamos:

MT_AFD:

	x		0		1..9, A..F		+		-		del	
→ 0		50	1	A		51	5	G	6	H	0	A
1	2	A		52		53		54		55		56
2		57	3	B	3	B		58		59		60
3	4	E	3	C	3	C	4	E	4	E	4	E
④												
⑤												
⑥												
	estado:	*acción:*										

La implementación sería:

```
ALex ()
{
  estado:= 0;
  LOOP until estado ≥ 4
  {
    acción:= MT_AFD.acción (estado, car);
    estado:= MT_AFD.estado (estado, car);

    IF estado = null then error (acción)
    else
      switch acción     /*ejecuta la acción semántica que corresponda*/
      {
        case A:  car:= leer ()
        case B:  valor:= valorHexadecimal (car)
                 car:= leer ()
        case C:  valor:= valor * 16 + valorHexadecimal (car)
                 car:= leer ()
        case E:  if valor < 2^31
                   then Return (Entero, valor)
                   else error (61)
        case G:  car:= leer (); Return (Más, -)
        case H:  car:= leer (); Return (Menos, -)
      }
  }
}
```