

Sintaxis y Semántica de los lenguajes.



CURSO: Z2102

AÑO 2025

PROFESOR: Roxana Leituz

TRABAJO PRÁCTICO N°: 2

TÍTULO: AUTÓMATAS

INTEGRANTES

Francesch, Valentina	222.315-6
Sanzo, Facundo	220.879-9
Julian Novoa	177.095-0
Bravo Arreyes, Matías	213.375-1

Instructivo:

Al ejecutar el programa, se le va a pedir elegir entre dos opciones, para elegir el modo.

```
Seleccione el modo:  
1. Reconocer constantes enteras (Punto 1)  
2. Evaluar operación aritmética (Punto 3)  
Opción: █
```

Si se selecciona la opción 1, se le va a pedir elegir entre dos posibles opciones para cargar la cadena a reconocer, insertar 1 para ingresar la cadena por línea de comandos, o 2 para leer un archivo, que debe llamarse “datos.txt” y debe encontrarse en la misma dirección que el archivo ejecutable del programa (el archivo debe contener únicamente la cadena, sin espacios ni saltos de línea), el programa luego devolverá los casos en los que se encontraron errores léxicos y por último la cantidad de números pertenecientes a cada grupo. Por último, pide presionar cualquier tecla y enter para finalizar.

```
Opción: 1  
Ingrese la cadena (números separados por #):  
343#123#20  
Cantidad de decimales: 3  
Cantidad de octales: 0  
Cantidad de hexadecimales: 0  
Presione ENTER para salir.  
  
...Program finished with exit code 0  
Press ENTER to exit console. █
```

Si se selecciona la opción 2, se le va a pedir elegir entre dos posibles opciones para cargar la cadena a reconocer, insertar 1 para ingresar la cadena por línea de comandos, o 2 para leer un archivo, que debe llamarse “datos.txt” y debe encontrarse en la misma dirección que el archivo ejecutable del programa (el archivo debe contener únicamente la cadena, sin espacios ni saltos de línea), el programa luego devolverá el resultado de la operación

```
Opción: 2  
Ingrese la expresión aritmética (ej: 3+4*7+3-8/4):  
3+5*3+4  
Resultado: 22  
Presione ENTER para salir.  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Especificaciones

Punto 1:

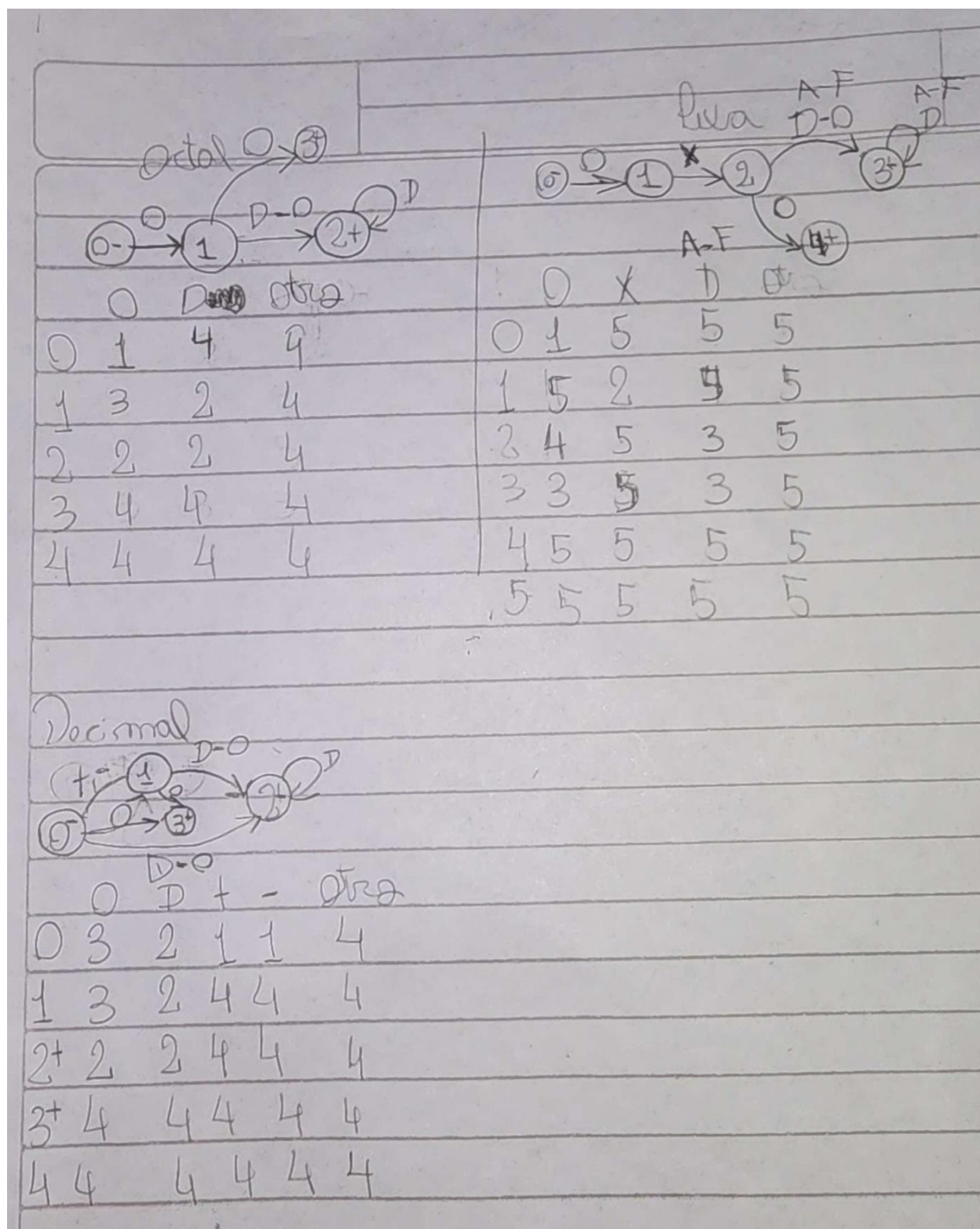
- A la hora de contemplar las diferentes formas de representar el 0 en cada base, diseñamos los autómatas de manera tal que el carácter '0', '+0' y '-0' sean reconocidos como caracteres decimales, el '00' sea reconocido como carácter octal y 0x0 como carácter hexadecimal, agregar más dígitos detrás en cualquiera de los tres casos invalida el valor.
- El grupo de hexadecimales reconocerá números sin importar si la 'x' se encuentra en minúscula o en mayúscula.
- Definimos arbitrariamente un largo máximo para la cadena de 100 caracteres, incluyendo el carácter '#'.

```

C automatas.c > ...
126 void verificar(char* cadena){
131 while(c != '\0'){
132     aux[j] = '0';
141
142     if (esDecimal(aux)) {
143         //printf("Este número es decimal: %s\n", aux);
144         contador[0] += 1;
145     }
146     else if (esOctal(aux)) {
147         //printf("Este número es octal: %s\n", aux);
148         contador[1] += 1;
149     }
150     else if (esHexa(aux)) {
151         //printf("Este número es hexadecimal: %s\n", aux);
152         contador[2] += 1;
153     }
154     else {
155         printf("El valor %s no aplica a ningun caso\n", aux);
156     }
157     //aux[j] = '\0'; // por seguridad termino el string
158     i++;
159     c = cadena[i];
160 }
161 printf("Cantidad de decimales: %d\n", contador[0]);
162 printf("Cantidad de octales: %d\n", contador[1]);
163 printf("Cantidad de hexadecimales: %d\n", contador[2]);
164 }
165
166
167 int main() {
168     char cadena[MAX];
169     int opcion;
170
171     printf("Elige la fuente de la cadena:\n");
172     printf("1. Ingresar por teclado\n");
173     printf("2. Leer desde archivo\n");
174     scanf("%d", &opcion);
175
176     if (opcion == 1) {
177         printf("Ingrese la cadena: ");
178         fgets(cadena, MAX, stdin);
179         verificar(cadena);
180     }
181     else if (opcion == 2) {
182         printf("Ingrese el nombre del archivo: ");
183         fgets(cadena, MAX, stdin);
184         verificar(cadena);
185     }
186 }

```

Elige la fuente de la cadena:
1. Ingresar por teclado
2. Leer desde archivo
Opcion: 2
Cadena a procesar: +0#-0#123#045#0x1F#99A#34256#03245#000000#0234#0x0234#+234#-4325#-0#0xac32
El valor 99A no aplica a ningun caso
El valor 000000 no aplica a ningun caso
El valor 0x0234 no aplica a ningun caso
Cantidad de decimales: 7
Cantidad de octales: 3
Cantidad de hexadecimales: 2
Presiona cualquier tecla y enter para salir.



Punto 2:

```
// Punto 2 -----
int charANumero(char c)
{
    return c - '0';
}
```

Aprovechamos el código ascii de los caracteres para encontrar el valor numérico real que vale.

Punto 3:

1. Cón esCcadenaValidaOperación() se valida que la cadena alterne entre números y operador, que no tenga operadores consecutivos, que no termine ni comience con un operador, y que sólo tenga los operadores +, -, * y / .
 - Se analiza carácter por carácter, usando la función esDecimal() del punto 1, y una nueva función auxiliar esOperador(), teniendo en cuenta la variable estado = 0 y estado = 1, que garantiza la alternancia de números y operadores.
2. Si la cadena es válida, se llama a evaluarOperación(), donde se convierte cada número detectado a int, con la función del punto 2 charANumero(), y se acumula su valor multiplicando por 10. Se construye un array numeros[] que contiene todos los operandos en orden.
3. Los operadores se guardan en orden en el array operadores[]
4. Se recorre el array de operadores[] en busca de el operador de multiplicación (*) o de división (/) y se resuelve con los operandos correspondientes. El resultado se vuelve a guardar en el array de numeros[], borrando lo innecesario (se corre todo una posición a la izquierda).
5. Se vuelve a recorrer el array de operadores[] en busca de sumas (+) o restas (-) y se realiza la cuenta con los números correspondientes.