UNIVERSIDAD RAFAEL LANDÍVAR FACULTAD DE INGENIERÍA LENGUAJES FORMALES Y AUTÓMATAS SECCIÓN 1 VESPERTINA MGTR. MOISES ALONZO

HOJA DE TRABAJO 3

Julio Anthony Engels Ruiz Coto 1284719

ER que reconoce si un número binario es par o impar.

```
Progran
                                               → % Program
                                                                                                → ® Main()
              class Program
                   0 references
                   static void Main()
                       while (true)
                           Console.WriteLine("Ingresa un número binario o escribe 'salir' para terminar:");
                           string binaryNumber = Console.ReadLine();
                           if (binaryNumber.ToLower() == "salir")
                               break;
                           Regex regex = new Regex(".*1$");
                           if (regex.IsMatch(binaryNumber))
                               Console.WriteLine("El número binario es impar.");
                           else
                               Console.WriteLine("El número binario es par.");
                           Console.WriteLine("Presiona cualquier tecla para continuar...");
                           Console.ReadKey();
                           Console.Clear();
```

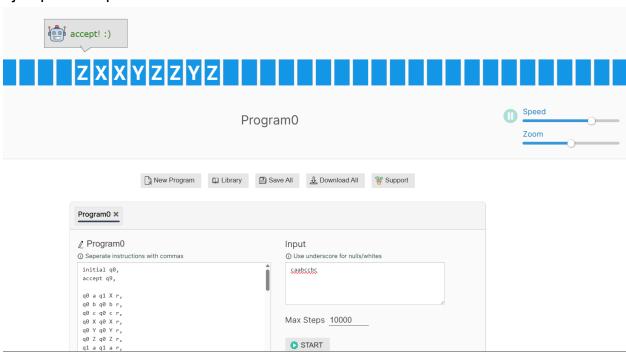
Con esta expresión regular ".*1\$" busca cualquier cadena que termine con el carácter '1'. El " .*" permite cualquier cantidad de caracteres antes del '1', pero el \$ asegura que el '1' debe ser el último carácter en la cadena.

```
© C:\Users\julio\Downloads\HT: × + ∨
Ingresa un número binario:
1010
El número binario es par.
```

```
© C:\Users\julio\Downloads\HT: × + ∨
Ingresa un número binario:
101011101
El número binario es impar.
```

MT que identifica que existan la misma cantidad de letras a y b, pero el doble de letras c

Ejemplo de input brindado en clase:



```
initial q0,
accept q9,
q0 a q1 X r,
q0 b q0 b r,
q0 c q0 c r,
q0 X q0 X r,
q0 Y q0 Y r,
```

- q0 Z q0 Z r,
- qlaqlar,
- q1 b q1 b r,
- q1 c q1 c r,
- q1 X q1 X r,
- q1 Y q1 Y r,
- q1 Z q1 Z r,
- q1 _ q2 _ 1,
- q2 b q3 Y l,
- q2 a q2 a l,
- q2 c q2 c l,
- q2 X q2 X 1,
- q2 Y q2 Y 1,
- q2 Z q2 Z l,
- q3 a q3 a l,
- q3 b q3 b l,
- q3 c q3 c l,
- q3 X q3 X l,
- q3 Y q3 Y l,
- q3 Z q3 Z l,
- q3 _ q4 _ r,
- q4 c q5 Z r,
- q4 a q4 a r,
- q4 b q4 b r,
- q4 X q4 X r,
- q4 Y q4 Y r,
- q4 Z q4 Z r,
- q5 c q6 Z r,
- q5 a q5 a r,
- q5 b q5 b r,
- q5 X q5 X r,
- q5 Y q5 Y r,
- q5 Z q5 Z r,
- q6 a q6 a r,
- q6 b q6 b r,
- q6 c q6 c r,
- q6 X q6 X r,
- q6 Y q6 Y r,
- q6 Z q6 Z r,
- q6_q7_1,

- q7 X q7 X l,
- q7 Y q7 Y l,
- q7 Z q7 Z l,
- q7 a q8 a l,
- q7 b q8 b l,
- q7 c q8 c l,
- q7_q9_r,
- q8 X q8 X l,
- q8 Y q8 Y l,
- q8 Z q8 Z l,
- q8 a q8 a l,
- q8 b q8 b l,
- q8 c q8 c l,
- $q8\,_\,q0\,_\,r$