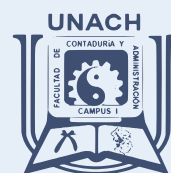


# ACTIVIDAD II

## COMPILADORES

Alumno: Alfredo López Méndez

Asesor: Dr. Luis Gutiérrez Alfaro



**Universidad Autónoma de Chiapas**  
Facultad de Contaduría y  
Administración Campus I  
Licenciatura en Ingeniería en  
Desarrollo y Tecnologías de Software  
Tuxtla Gutiérrez, Chiapas

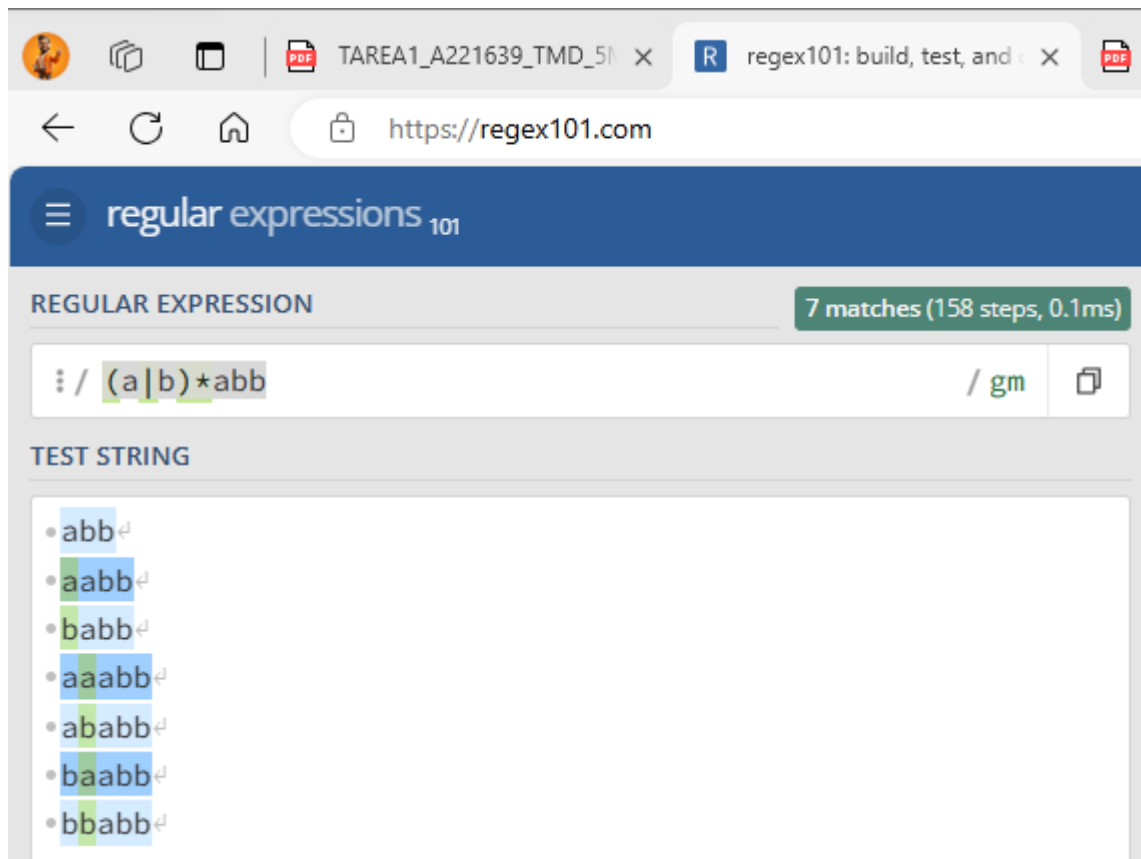
5to Semestre; Grupo M  
Entrega:  
jueves, 15 de agosto de 2024

# Actividad II. Ejercicios

## Compiladores

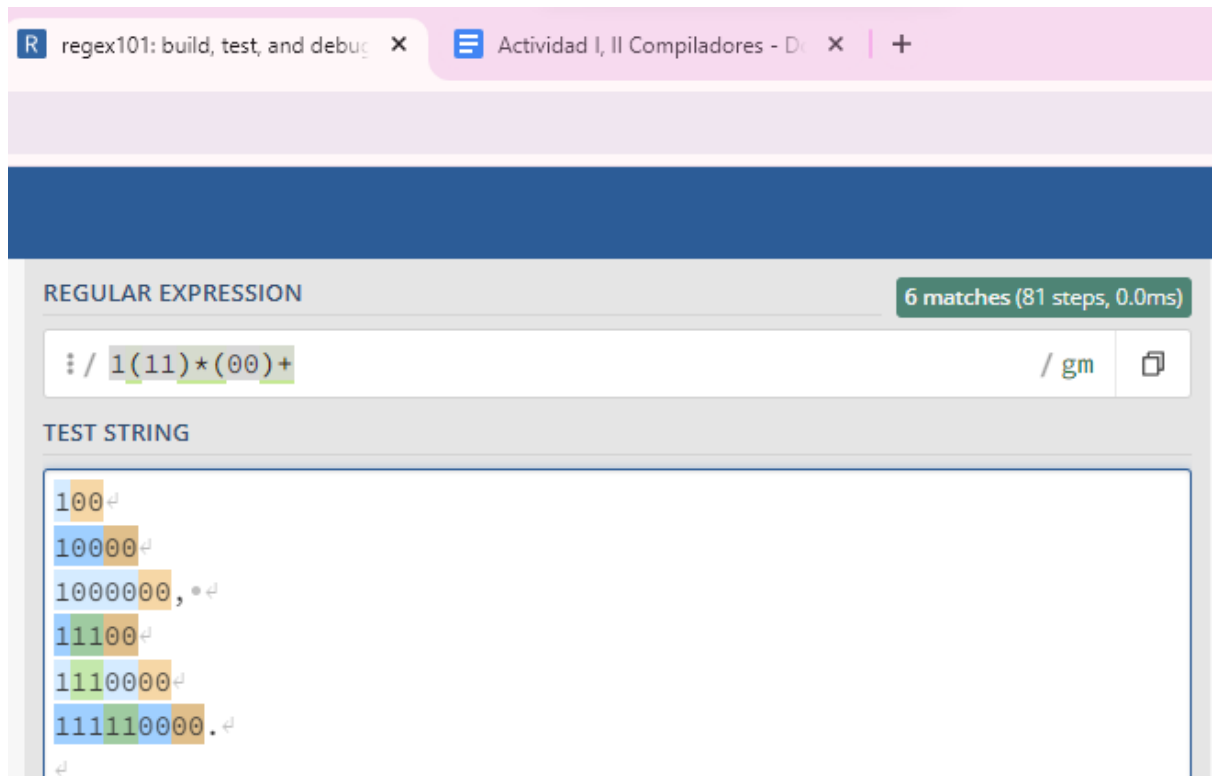
- 1) Realice una expresión regular de todas las cadenas con símbolos a y b, que terminan con el sufijo abb. Ejemplo de estas cadenas son: abb, aabb, babb, aaabb, ababb, baabb, bbabb.

La expresión regular quedaría como: `(a|b)*abb`



- 2) Realice una expresión regular de todas las cadenas con símbolos 0 y 1, que primero tengan los símbolos 1's con longitud impar y después aparezcan los 0's con longitud par. Ejemplo de éstas cadenas son: 100, 10000, 1000000, 11100, 1110000, 111110000.

La expresión regular quedaría como: `1(11)*(00)+`



3) Para la expresión regular `(+|-)?d + .d +` indique las cadenas correctas de los siguientes incisos. (Nota. En esta expresión él . es un símbolo no el operador concatenación y d representa los dígitos del 0 al 9)

- a) `-20.43`
- b) `0.3216`
- c) `329.`
- d) `217.92`
- e) `+2019`
- f) `+.762`
- g) `-.4555`

4) Obtenga un AFD dado el siguiente lenguaje definido en el alfabeto  $\Sigma = \{a, b, c\}$ . El Conjunto de cadenas que inician en la sub cadena "ac" y terminan en la sub-cadena "ab"

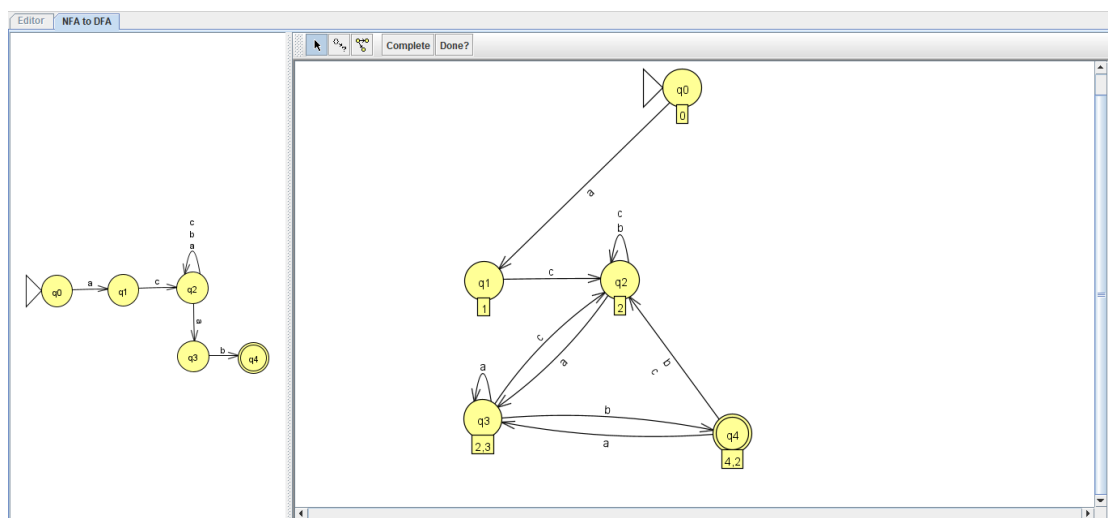
Editor Multiple Run

Table Text Size

Input	Result
acbbab	Accept
cabab	Reject
cacabc	Reject
abac	Reject
acabab	Accept
acacab	Accept
accba	Reject

```

graph LR
    q0((q0)) -- a --> q1((q1))
    q1 -- c --> q2((q2))
    q2 -- c --> q2
    q2 -- b --> q3((q3))
    q3 -- b --> q4(((q4)))
  
```



- 5) Obtenga un AFND dado el siguiente lenguaje definido en el alfabeto  $\Sigma = \{a, b, c\}$ . El Conjunto de cadenas que no inician en la sub cadena "ac" o no terminan en la sub-cadena "ab".

Editor

Multiple Run

```

graph LR
    q0((q0)) -- c --> q1((q1))
    q0 -- b --> q2((q2))
    q1 -- a --> q2
    q1 -- c --> q3((q3))
    q2 -- c --> q3
    q2 -- a --> q4((q4))
    q3 -- a --> q1
    q3 -- b --> q2
    q3 -- a --> q4
    q4 -- b --> q3
  
```

Table Text Size

Input	Result
acbbab	Reject
cabab	Reject
cacabc	Accept
abac	Accept
acabab	Reject
acacab	Reject
accca	Reject
aabcca	Accept
bababa	Accept
caacba	Accept
acabac	Reject