



TRABAJO PRÁCTICO N° 2
CONSTRUCCIÓN DE THOMPSON Y CONVERSIÓN DE AFN EN AFD
IMPLEMENTACIÓN CON PYTHON PARA VALIDACIÓN DE DATOS
MEDIANTE EXPRESIONES REGULARES

Integrantes:

- Mancuso, Augusto
- Narvaes, Franco
- Kark Verstraete, Augusto

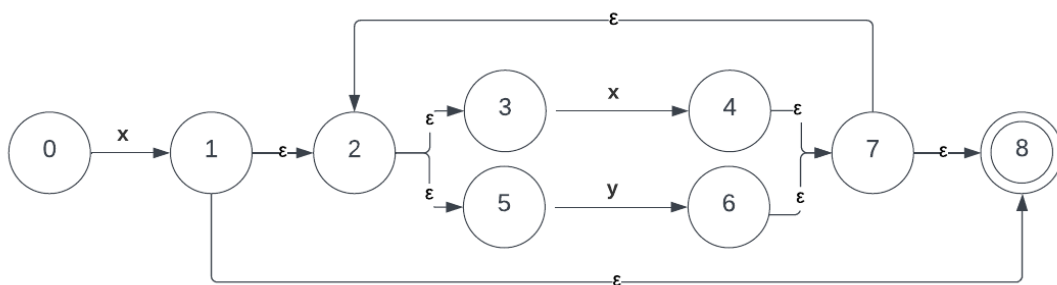
PARTE A

Conceptos fundamentales: autómatas finitos deterministas y no deterministas, aceptación de cadenas, obtención de autómatas a partir de expresiones regulares, y conversión de AFN en AFD

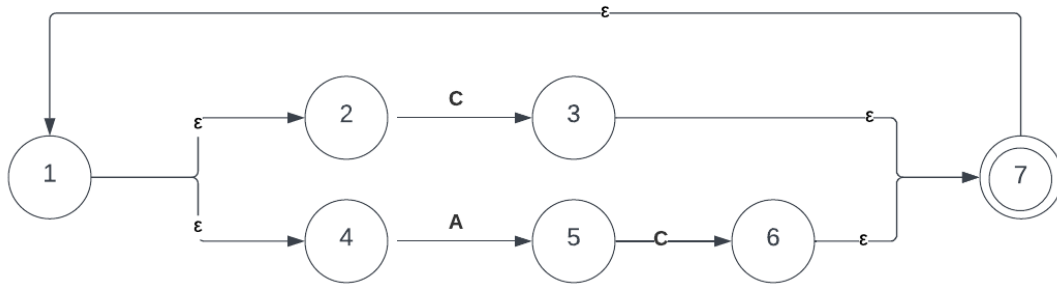
Ejercicio 1:

Cree, mediante “Construcción de Thompson”, los siguientes autómatas finitos no deterministas:

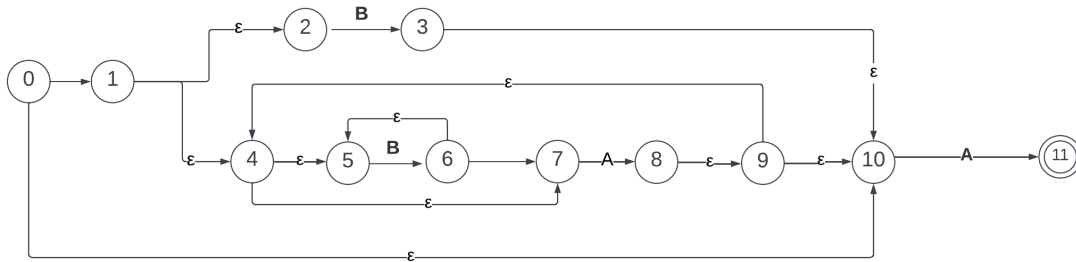
a) Que reconozca la expresión regular $x (x | y)^*$.



b) Que reconozca la expresión regular $(c | a c)^*$.



c) Que reconozca la expresión regular $(b \mid (b^* a)^*)a$.



Ejercicio 2:

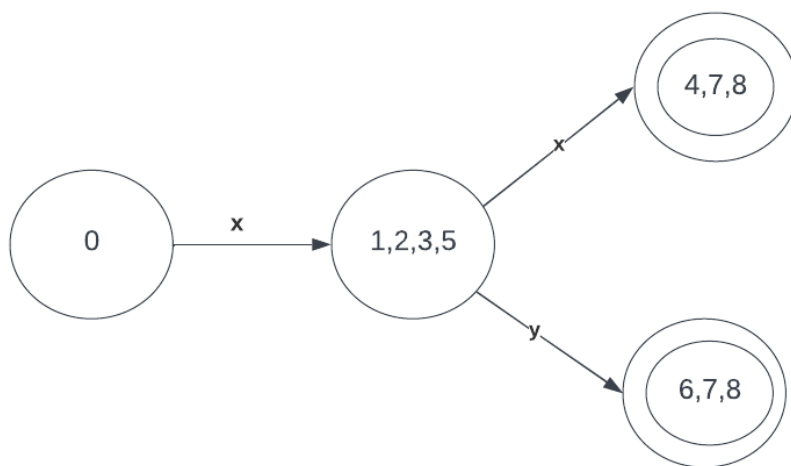
Convierta, utilizando construcción de subconjuntos, los autómatas finitos no deterministas realizados en el ejercicio anterior, en autómatas finitos deterministas.

a)

$0 = \{0, 1\}$
 $\{0, 1\}x = \{1\}$
 $\{0, 1\}y = \{\}$

$1 = \{1, 2, 3, 5, 8\}$
 $\{1, 2, 3, 5, 8\}x = \{4\}$
 $\{1, 2, 3, 5, 8\}y = \{6\}$

$2 = \{2, 3, 5, 7, 8\}$
 $\{2, 3, 5, 7, 8\}x = \{4\}$
 $\{2, 3, 5, 7, 8\}y = \{6\}$



b)

$1 = \{1, 2, 4\}$

$\{1, 2, 4\}a = \{5\}$

$\{1, 2, 4\}c = \{3\}$

$\{5\} = \{5, 6\}$

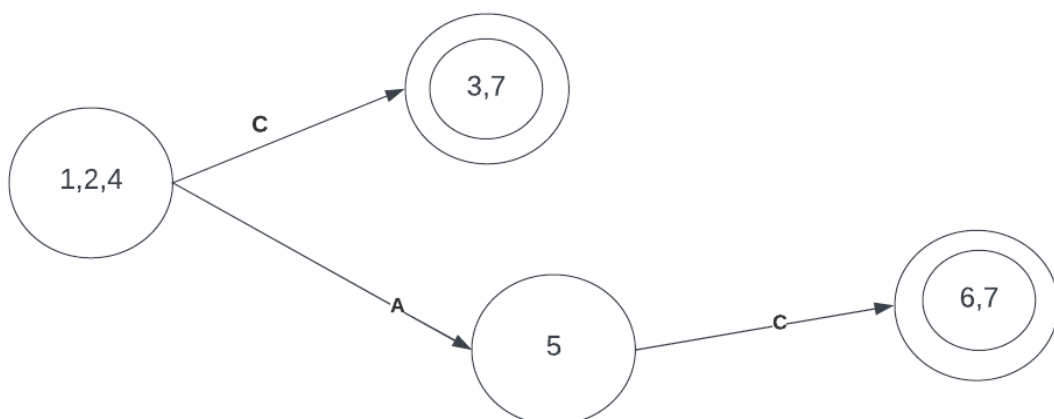
$\{3\} = \{1, 2, 3, 4, 7\}$

$\{5, 6\}a = \{\}$

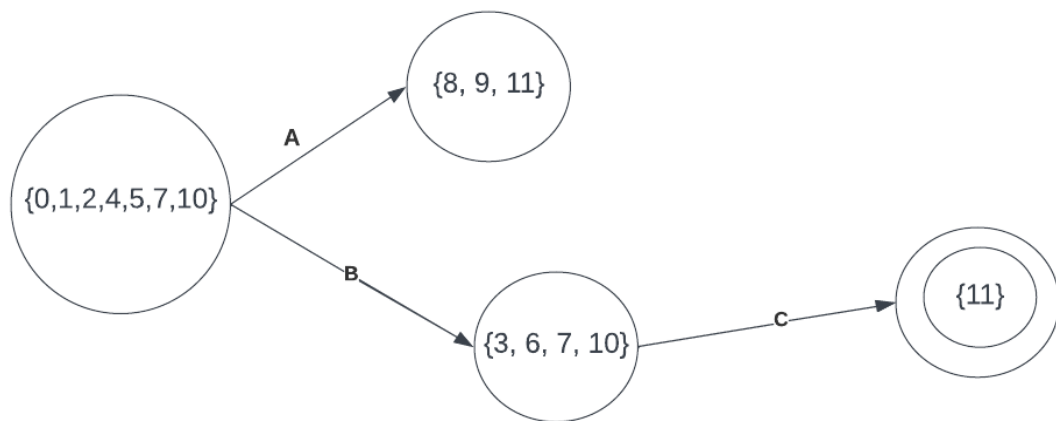
$\{5, 6\}c = \{6\}$

$\{1, 2, 3, 4, 7\}a = \{5\}$

$\{1, 2, 3, 4, 7\}c = \{3\}$



c)



$0 = \{0,1, 2, 4, 5, 7,10\}$

$\{0,1, 2, 4, 5, 7, 10\}a = \{8, 11\}$

$\{0,1, 2, 4, 5, 7, 10\}b = \{3, 6\}$

$\{3, 6\} = \{3, 6, 7, 10\}$

$\{3, 6, 7, 10\}a = \{11\}$

$\{3, 6, 7, 10\}b = \{\}$

$[8,11]=\{8,11,9\}$

$\{8,11,9\}a = \{\}$

$\{8,11,9\}b = \{\}$

$\{11\} = \{ 11\}$

$\{11\}a = \{\}$

$\{11\}b = \{\}$

PARTE B

Conceptos fundamentales: Expresiones regulares y definiciones regulares.

Ejercicio 1

Escriba las descripciones para los lenguajes generados por las siguientes expresiones regulares:

1. $(ca|b)^*(a|cc)^*$: todas las cadenas que contienen la subcadena "ca" o "b", seguida la subcadena "a" o "cc". Por ejemplo, algunas cadenas en este lenguaje son: "b", "caa", "caaca", "bacc", "bca", "baccacc".

2. **(0|1|..|9|A|B|C|D|E|F)(w|W)** : consiste en todas las cadenas que comienzan con un dígito decimal o letra mayúscula (A, B, C, D, E, o F), seguido de un carácter "w" o "W". Por ejemplo, algunas cadenas en este lenguaje son: "0w", "1W", "AFw", "9W", "Bbw".
3. **(A|B|...|Z)(a|b|...z)*** : consiste en las cadenas que comienzan con una letra mayúscula del alfabeto (A a Z), seguida de cualquier cantidad de letras minúsculas del alfabeto (a a z). Por ejemplo, algunas cadenas en este lenguaje son: "A", "Bx", "Cdefg", "Zyxwvu", "Yz".

Ejercicio 2

Escriba las expresiones regulares para validar:

1. Fecha con formato dd/mm/yyyy o dd-mm-yyyy

`^(0[1-9]|[1-2][0-9]|3[0-1])(\|)(0[1-9]|1[0-2])(\|)\d{4}$`

2. Número real con dos decimales y separador de miles.

`^-?(?:(?:\d{1,3},)*\d{1,3}|(?:\d+)?(?:\.\d{2})?)?$`

3. ID de un vídeo de Youtube

`^[a-zA-Z0-9_-]{11}$`

4. Cuenta de Email de alumno de la Universidad de Mendoza

`^[a-z]{2}\.[a-z]+@(alumno\.um\.edu\.ar)$`

5. Número de teléfono móvil de Argentina, que incluya código de país, de provincia, y el 15.

`^54[9](11|[2368])[1-9]15[0-9]{7}$`

6. CUIL.

`^[0-9]{2}(-)[0-9]{8}(-)[0-9]{1}$`

7. Seguridad de una contraseña, que incluya:

- Que contenga al menos un número.
- Que contenga al menos una letra mayúscula.
- Que contenga al menos carácter especial.
- La longitud sea como mínimo 8 caracteres.
- La longitud máxima no debe mayor a 16 caracteres.

`^(?=.*\d)(?=.*[A-Z])(?=.*[!@#$%^&*()-_+=])(?!.*\s){8,16}$`

Ejercicio 3:

Utilizando el lenguaje de programación Python, implementar la validación de las expresiones regulares del ejercicio anterior (Ejercicio 2)

Ejemplo: validación de extensiones para imágenes.

```
import re
regex = re.compile(r"jpg|png|gif|bmp|svg")
img_ext = input("Ingrese una extensión de una imagen: ")
if regex.match(img_ext):
    print('La extensión ', img_ext, 'se corresponde con la extensión de una imagen')
else:
    print('La extensión ', img_ext, 'no se corresponde con la extensión de una')
```

imagen

```
import re
print('Qué expresión regular desea verificar?')
print('1: validación de fecha\n'
      '2: validación de número real\n'
      '3: validación de ID de video de Youtube\n'
      '4: validación de mail de la universidad\n'
      '5: validación de número de celular argentino\n'
      '6: validación de CUIL\n'
      '7: validación de contraseña')

opcion = int(input('Ingrese el número de la opción: '))

def validacion_fecha(fecha):
    patron =
re.compile(r'^([0-9]|[1-2][0-9]|3[0-1]) (\/|-) ([0-9]|1[0-2]) (\/|-) \d{4}$')

    if patron.match(fecha):
        print('La fecha ingresada es correcta')
    else:
        print('La fecha ingresada es incorrecta')

def num_real(numero):
    patron=re.compile(r'^-?(?:\d{1,3},)*\d{1,3}|\d+)(?:\.\d{2})?$')
    if patron.match(numero):
        print('El número ingresado es correcto')
    else:
        print('El numero ingresado es incorrecto')

def video_youtube(string):
    patron=re.compile(r'^[a-zA-Z0-9_-]{11}$')
    if patron.match(string):
        print('El ID ingresado es correcto')
    else:
        print('El ID ingresado es incorrecto')

def mail_uni(mail):
```

```

patron=re.compile(r'^[a-z]{2}\.[a-z]+@(alumno\.um\.edu\.ar)$')
if patron.match(mail):
    print('El mail ingresado es correcto')
else:
    print('El mail ingresado es incorrecto')

def telefono_arg(cel):
    patron=re.compile(r'^54[9](11|[2368][1-9])15[0-9]{7}$')
    if patron.match(cel):
        print('El celular ingresado es correcto')
    else:
        print('El celular es incorrecto')

def Cuil(cuil_usr):
    patron=re.compile(r'[0-9]{2}(-)[0-9]{8}(-)[0-9]{1}$')
    if patron.match(cuil_usr):
        print('El CUIL ingresado es correcto')
    else:
        print('El CUIL es incorrecto')

def password(contraseña):
    patron=re.compile(r'^(?=.*\d)(?=.*[A-Z])(?=.*[!@#$%^&*()-_+=])(?!.*\s){8,16}$')
    if patron.match(contraseña):
        print('La contraseña ingresada es correcta')
    else:
        print('La contraseña ingresada es incorrecta')

if opcion == 1:
    fecha = input('Ingrese una fecha (dd/mm/aaaa) o (dd-mm-aaaa): ')
    verificacion=validacion_fecha(fecha)
elif opcion == 2:
    numero = input('Ingrese un número real con dos decimales(punto) y separador de miles(coma) positivo o negativo ')

```

```
    verificacion1=num_real(numero)
elif opcion == 3:
    string=input('Ingrese el ID de Youtube')
    verificacion2=video_youtube(string)
elif opcion == 4:
    mail=input('Ingrese su mail de la universidad')
    verificacion3=mail_uni(mail)
elif opcion == 5:
    cel=input('Ingrese el numero de celular con código de pais (54), de
provincia (ej 11 BsAS, 26 Mendoza) seguido de 15(numero de celular) ')
    verificacion4=telefono_arg(cel)
elif opcion == 6:
    cuil_usr=input('Ingrese el cuil (nn-dni-verficicador): ')
    verificacion5=Cuil(cuil_usr)
elif opcion == 7:
    contraseña=input()
    verificacion5=password(contraseña)
else:
    print('Opción incorrecta')
```