Unidad Profesional Interdisciplinaria en
Ingeniería y Tecnologías Avanzadas

## Departamental 1

## Práctica 1.1


Nombre completo:

**José Ángel Abarca Romero**


2TM9 - Lógica Difusa

Profesora: Dra. Yesenia Eleonor González Navarro

Academia de Sistemas

30 de marzo de 2023

# Índice

# 1. Introducción

La teoría de los conjuntos difusos nos ayuda a desarrollar sistemas que describan el comportamiento de cierto que fenómeno cuyas características intrínsecas no permiten el uso de sistemas basados en lógica tradicional o cuyo rendimiento se ve incrementado al usar universos difusos como variables de entrada.

Como es de esperarse, a pesar de que las variables de entrada y valores de salida en un sistema de inferencia difuso son universos difusos, en realidad los elementos controlados por dichos sistemas requieren de valores certeros que les indiquen qué hacer en un momento determinado dependiendo de la condiciones a la entrada del sistema. Es por ello que surge la necesidad de desarrollar métodos que permitan difusificar valores certeros para poder ser interpretados por el sistema difus, y a su vez, métodos que permitan desdifusificar valores difusos en valores certeros que al final son los que determinan el comportamiento del elemento o fenómeno del cual estamos describiendo su comportamiento.

Existen diversos tipos de sistemas de inferencia que cumplen con lo mencionado en el párrafo anterior, entre ellos se encuentran los sistemas de inferencia difusa de tipo Mamdani. En ésta práctica, estaré desarrollando un programa que implementa un sistema de inferencia difusa de tipo Mamdani para el problema que se resuelve en el artículo Çómo diseñar un sistema de inferencia de tipo Mamdani".

# 2. Marco Teórico

## 2.1. Sistemas de Inferencia de Tipo Mamdani

Estos sistemas de inferencia utilizan universos difusos tanto en su entrada como en su salida. Existen diversos métodos de desdifusificación entre los cuales se encuentran: izquierda del máximo, media del máximo, derecha del máximo, bicéctriz, promedio de pesos y centroide del área.

El método del centroide del área resulta ser el de menor complejidad de todos y aún así se obtienen resultados lo suficientemente aproximados a otros métodos para considerarlo correcto. Su ecuación se muestra a continuación

$$z^* = \frac{\sum \mu_{CT}(z) z}{\sum \mu_{CT}(z)} \tag{1}$$

donde $z^*$ es el valor certero desdifusificado, que resulta de evaluar cada elemento z del universo de salida ponderado por su pertenencia en el conjunto de salida $CT$.

# 3. Desarrollo

## 3.1. Instrucciones

Implemente en Python un sistema de inferencia de tipo Mamdani para el problema que se resuelve en el artículo Çómo diseñar un sistema de inferencia de tipo Mamdani"(documento cargado en Archivos). Realice 2 códigos:

1. Un código que genere las gráficas de los universos de entrada y salida caracterizados, los conjuntos de salida recortados, la curva resultante de unir los conjuntos de salida recortados y el valor desdifusificado de la variable de salida $z^*$.

2. Un código que genere la superficie de control para la variable de salida.

### 3.2. Ejercicio 1

Comencé definiendo el rango de operación para los universos de entrada y de salida. A partir de ello obtuve los conjuntos definidos en todos los universos mediante las expresiones mostradas en la Figura 1

$$A1 = \frac{1}{1+exp(0.3(x-50))} \quad , \quad A2 = \frac{1}{1+exp(-0.3(x-50))}$$
$$B1 = exp^{-\frac{1}{2}\left(\frac{y-25}{20}\right)^2} \quad , \quad B2 = exp^{-\frac{1}{2}\left(\frac{y-75}{20}\right)^2}$$
$$C1 = \frac{1}{1+exp(0.3(z-50))} \quad , \quad C2 = exp^{-\frac{1}{2}\left(\frac{z-75}{20}\right)^2}$$

Figura 1: Ecuaciones que definen a los conjutos de los universos de entrada y salida.

Luego grafiqué los conjuntos obtenidos (Ver Figuras 3, 4 y 5 en la sección de resultados). Posteriormente, le pido al usuario que ingrese el par de valores $x$ y $y$ a la entrada del sistema para comenzar la difusificación.

Primero, obtengo los grados de pertenencia del los valores de entrada a los conjuntos difusos en sus respectivos universos de acuerdo a las expresiones de la Figura 1. Con los valores que obtuve evalué las reglas "si-entonces"mediante composición difusa de tipo max - min para obtener la agregación hacia los conjuntos C1 y C2 del universo Z de acuerdo a la tabla de inferencia mostrada a continuación.

Cuadro 1: Tabla de inferencia

| **X/Y** | A1 | A2 |
|---|---|---|
| **B1** | C1 | C2 |
| **B2** | C1 | C2 |

Con los valores resultantes de la agregación realicé el corte de los conjuntos C1 y C2 (ver Figura 6). Con estos dos conjuntos nuevos creé una nueva curva llamada $CT$ (Figura 7), la cual utilizo para obtener una orden certera mediante la desdifusificación por el método del centroide de área, cuya ecuación definí previamente (Ecuación 1).

El valor de $z^*$ depende de los valores $x$ y $y$ ingresados por el usuario. En la sección de resultados se puede ver el valor de $z^*$ para $x = 60$ y $y = 35$, valores propuestos en el artículo *Cómo diseñar un sistema de inferencia de tipo Mamdani* [1].

### 3.3. Ejercicio 2

El ejercicio 2 resulta ser una extensión de lo realizado en el ejercicio 1, pues es necesario realizar el mismo procedimiento pero ahora se evalúan todos los pares de entrada $x$ y $y$ posibles.

Esto se realiza mediante dos ciclos de tipo *for* anidados, los cuales recorren los vectores $x$ y $y$ definidos previamente. Esto requiere de un tiempo considerable de compilación pues a medida que aumenta la resolución de los vectores utilizados, también se incrementa el número de operaciones necesarias para realizar todos los cálculos.

## 4. Resultados

```
Ingrese el valor de x: 60
Ingrese el valor de y: 35
Tabla de inferencia:  [[0.04742587317756678, 0.8824969025845955], [0.04742587317756678, 0.1353352832366127]]
z* =  69.5526846877605
```

Figura 2: Tabla de inferencia y valor desdifusificado para $x = 60$ y $y = 35$

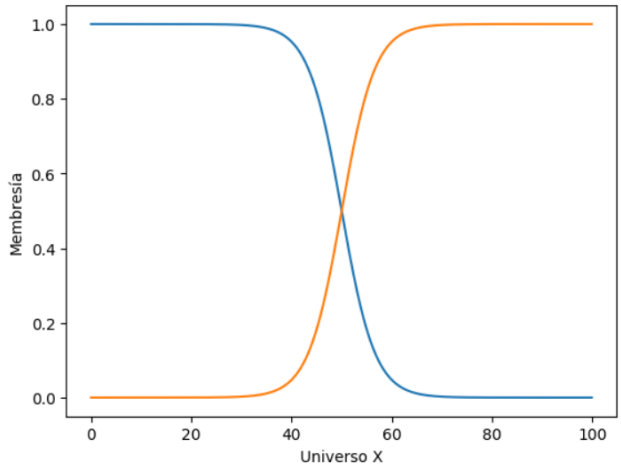


Figura 3: Conjuntos A1 y A2 definidos en X

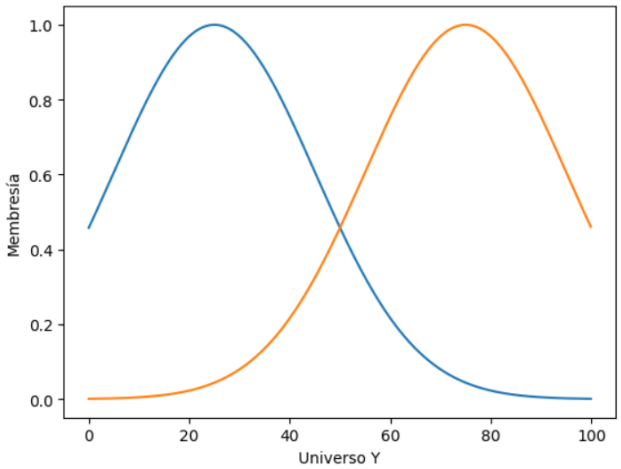


Figura 4: Conjuntos B1 y B2 definidos en Y

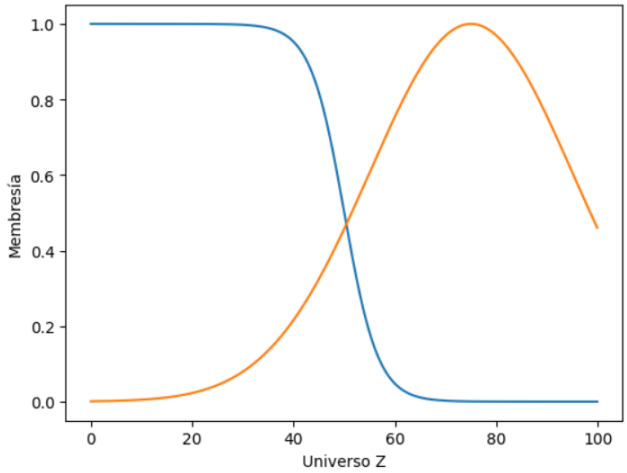


Figura 5: Conjuntos C1 y C2 definidos en Z

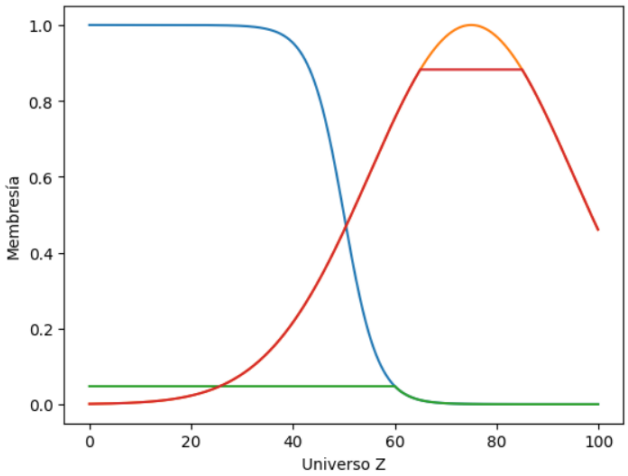


Figura 6: Conjuntos C1 y C2 vs sus versiones recortadas

## 5. Conclusiones

Se obtuvieron los resultados esperados tanto para el ejercicio 1 como para el ejercicio 2. Si bien, con el desarrollo de esta práctica quedó bien ilustrado los metodos usados para un sistema de inferencia difusa de tipo Mamdani, soy conciente de que no es el único método para obtener un valor certero a partir de cierto número de universos de entrada difuso. Además, considero que el tiempo de ejecución necesario para realizar el ejercicio 2 fue obsesivo, por lo que para programas futuros considero pertinente atacar los ejercicios desde otra perspectiva para aminorar la carga computacional que requiere.

## Referencias

[1] B. Granados Rojas P. N. Cortez Herrera Y. E. González Navarro. *Cómo Diseñar un Sistema de Inferencia Difusa de Tipo Mamdani*. 1st. La Laguna Ticomán, CDMX, MX: Boletín UPIITA, 2019.

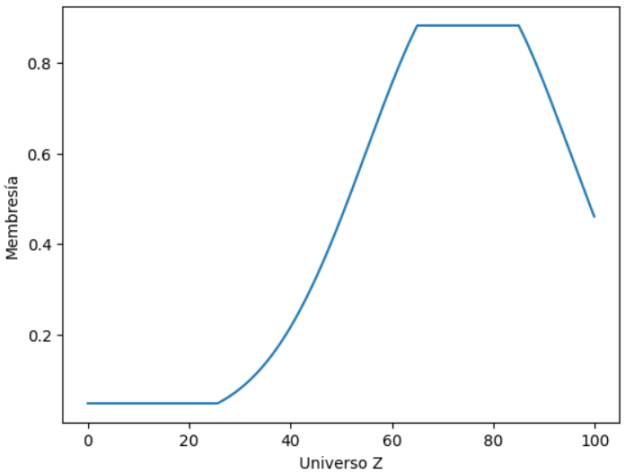


Figura 7: Conjunto CT

# Apéndice A Códigos en Python

```
import numpy as np
from math import exp
import matplotlib.pyplot as plt
from matplotlib import cm


x = np.arange(0, 100, 0.1)
y = np.arange(0, 100, 0.1)
z = np.arange(0, 100, 0.1)
X, Y = np.meshgrid(x, y)

A1 = np.zeros(len(x))
A2 = np.zeros(len(x))
B1 = np.zeros(len(y))
B2 = np.zeros(len(y))
C1 = np.zeros(len(z))
C2 = np.zeros(len(z))
CZ = np.zeros((len(x),len(y)))


#Obtención de los grados de pertenencia a los conjuntos de los universos de entrada
def difusificacion():
  valx = int(input("Ingrese el valor de x: "))
  memA = [A1[valx*10],A2[valx*10]]
  valy = int(input("Ingrese el valor de y: "))
  memB = [B1[valy*10],B2[valy*10]]
  composicion(memA,memB)

#Composición max - min (Evaluación de las reglas "si-entonces")
def composicion(memA,memB):

#Tabla de inferencia (intersección - min)
  tabla = [[np.minimum(memA[0],memB[0]),np.minimum(memA[1],memB[0])],
           [np.minimum(memA[0],memB[1]),np.minimum(memA[1],memB[1])]]

  print("Tabla de inferencia: ",str(tabla))
  agregacionC1 = max(tabla[0][0],tabla[1][0]) #Agregación - max
  agregacionC2 = max(tabla[0][1],tabla[1][1])
  agregacion(agregacionC1,agregacionC2)

def agregacion(agregacionC1,agregacionC2):
  aux1 = agregacionC1*np.ones(len(C1))
  aux2 = agregacionC2*np.ones(len(C2))

  C1_r = np.minimum(C1,aux1)
  C2_r = np.minimum(C2,aux2)

  plt.figure(4)
  plt.plot(z,C1,z,C2,z,C1_r,z,C2_r)
  plt.xlabel("Universo Z")
  plt.ylabel("Membresía")

  dedifusificacion(C1_r,C2_r)

def dedifusificacion(C1_r,C2_r):
  z_d = 0
  aux1 = 0
  aux2 = 0
  CT = np.maximum(C1_r,C2_r)
  plt.figure(5)
  plt.plot(z,CT)
```

```
# Para la ejecución de este programa es necesario contar con las variables
# declaradas en el ejercicio 1

def difusificacion(valx,valy):
  memA = [A1[valx],A2[valx]]
  memB = [B1[valy],B2[valy]]
  return composicion(memA,memB)

def composicion(memA,memB):

  tabla = [[np.minimum(memA[0],memB[0]),np.minimum(memA[1],memB[0])],
           [np.minimum(memA[0],memB[1]),np.minimum(memA[1],memB[1])]]
  agregacionC1 = max(tabla[0][0],tabla[1][0])
  agregacionC2 = max(tabla[0][1],tabla[1][1])
  return agregacion(agregacionC1,agregacionC2)


def agregacion(agregacionC1,agregacionC2):
  aux1 = agregacionC1*np.ones(len(C1))
  aux2 = agregacionC2*np.ones(len(C2))

  C1_r = np.minimum(C1,aux1)
  C2_r = np.minimum(C2,aux2)

  return dedifusificacion(C1_r,C2_r)

def dedifusificacion(C1_r,C2_r):
  z_d = 0
  aux1 = 0
  aux2 = 0
  CT = np.maximum(C1_r,C2_r)

  for i in range(len(z)):
    aux1 += CT[i]*i/10
    aux2 += CT[i]

  z_d = aux1/aux2
  return z_d

def malla():
  for i in range(len(x)):
    for j in range(len(y)):
      CM[i,j] = difusificacion(i,j) #Superficie de control generada
  fig, ax = plt.subplots(subplot_kw={"projection": "3d"})
  ax.set_xlabel('Y')
  ax.set_ylabel('X')
  ax.set_zlabel('Z')
  surf = ax.plot_surface(X, Y, CM, cmap=cm.coolwarm,
                         linewidth=0, antialiased=False)


malla()
```

Listing 2: Código del ejercicio 2.

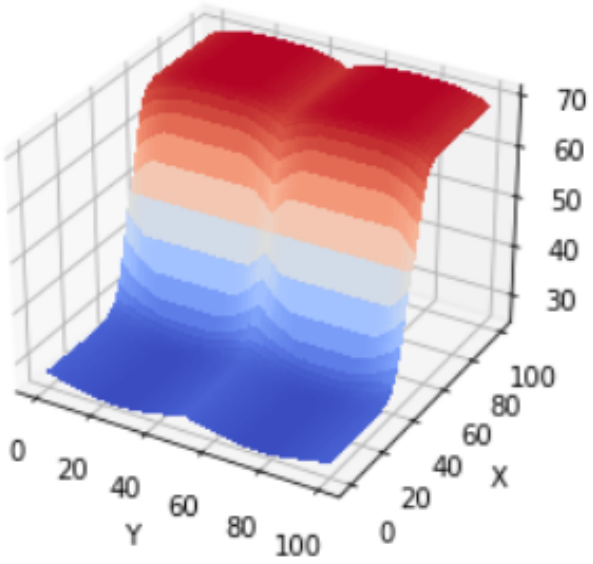


Figura 8: Superficie de control resultante