

# PROYECTO#2 DE SIMULACIÓN LÓGICA DIFUSA CURSO 2016-2017

Amelia Rabanillo Echaniz<sup>1</sup>

<sup>1</sup> C412, a.rabanillo@estudiantes.matcom.uh.cu

## 1. DETALLES DE IMPLEMENTACION

Para definir un sistema de lógica difusa en la implementación brindada se cuenta con la clase *fuzzy\_system* que para crearla recibe como parámetro un array de reglas, un array con las variables de entrada del sistema y la variable de salida.

Para definir las variables del sistema se cuenta con la clase *characteristic* que recibe un string, el cual es el nombre lingüístico de la variable en cuestión. La clase *characteristic* permite agregar funciones de pertenencia para cada estado de la variable mediante los métodos *add\_trapezoidal*, *add\_triangular* y *add\_regular*, los cuales reciben un string para identificar el estado y un array de puntos que representan la función en cada caso.

Además se pueden representar condiciones mediante la clase *condition* la cual recibe un array de tuplas, donde el primer elemento de la tupla es una característica y el segundo alguna función de pertenencia agregada a esa característica, también recibe un array con los operadores a aplicar entre cada una de las tuplas.

Las clases y métodos anteriores permiten crear las reglas que necesita *fuzzy\_system*. Estas reglas reciben dos condiciones, la primera representando la parte izquierda de la implicación y la segunda la parte derecha.

Para que el sistema difuso devuelva un valor anteriormente deben llamarse al método *set\_value* de cada variable con un parámetro numérico y luego llamar al método *Apply* del sistema difuso el cual devuelve un valor numérico para la variable designada como de salida y teniendo en cuenta los valores aplicados en el *set\_value* de cada variable.

El sistema resuelve el problema usando como método de fusificación mamdani y de defusificación el método del primer máximo, donde del polígono que devuelve el método de mamdani se escoge el punto que brinde la evaluación máxima con menor

valor de x.

## 2. EJEMPLOS

En la implementación se incluyen dos ejemplos que se detallan a continuación

Ejemplo#1:

Las variables involucradas son aire (frio, fresco, correcto, cálido, caliente) y velocidad (parada, lenta, media, rápida, máxima) y las reglas son:

R1: aire = frio => velocidad = parada

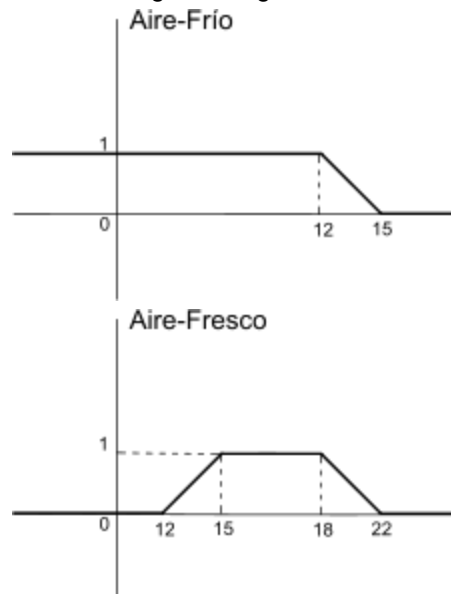
R2: aire=fresco => velocidad = lenta

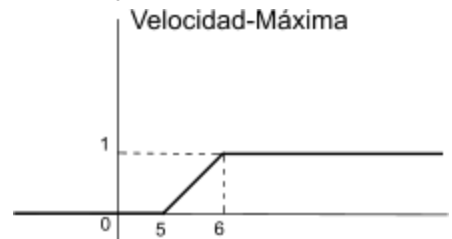
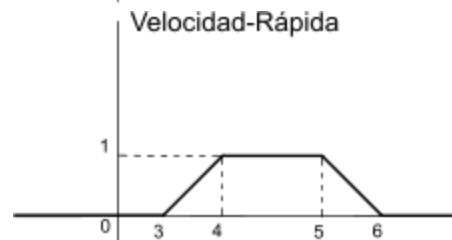
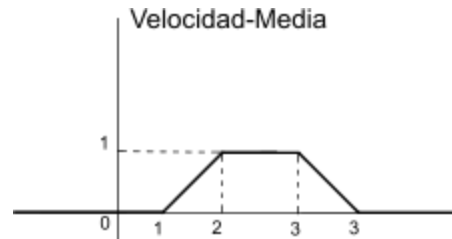
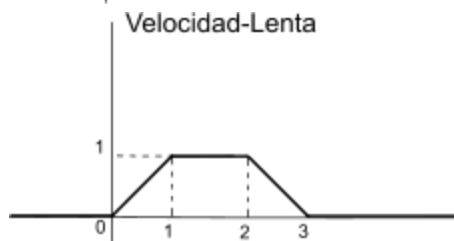
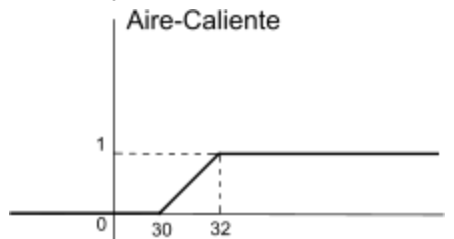
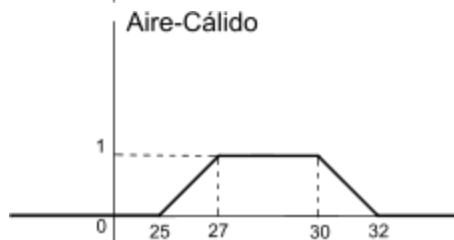
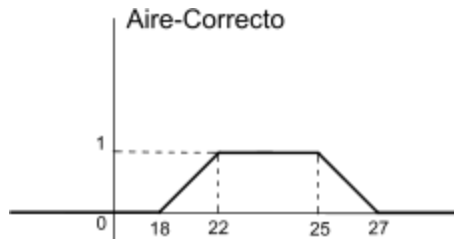
R3: aire=correcto => velocidad = media

R4: aire = cálido => velocidad = rápida

R5: aire =caliente => velocidad =máxima

Las funciones para cada característica corresponden a los siguientes gráficos:





Para este ejemplo se usaron 5 casos de prueba que devolvieron la siguiente salida:

Ejemplo#1

```

Temperatura del aire: 0, velocidad: 0
Temperatura del aire: 15, velocidad: 1
Temperatura del aire: 25, velocidad: 2
Temperatura del aire: 30, velocidad: 4
Temperatura del aire: 45, velocidad: 6

```

Ejemplo#2:

Para este ejemplo se tienen las variables servicio (pobre, bueno, excelente), comida (mala, regular, deliciosa) y como variable de salida se tiene propina (poca, promedio, generosa).

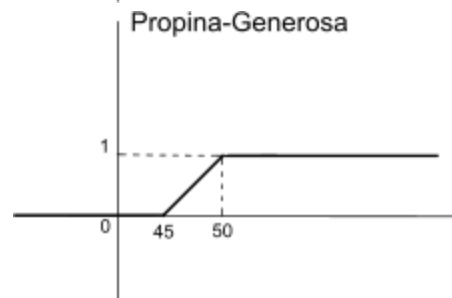
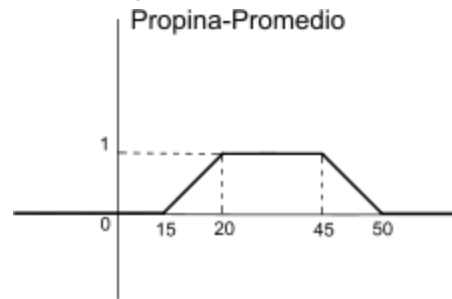
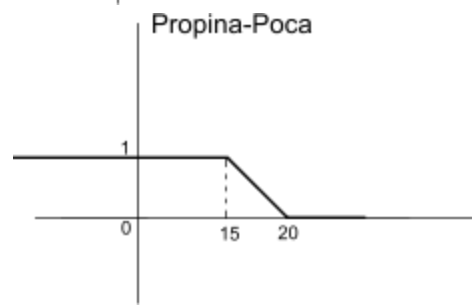
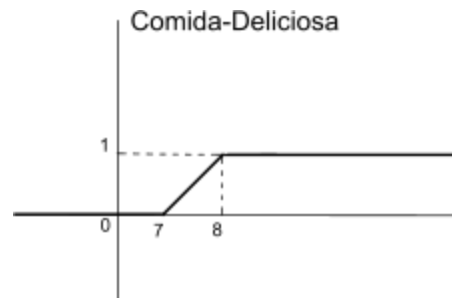
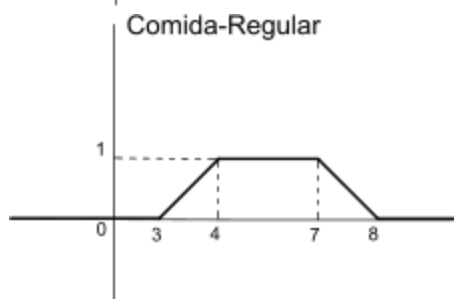
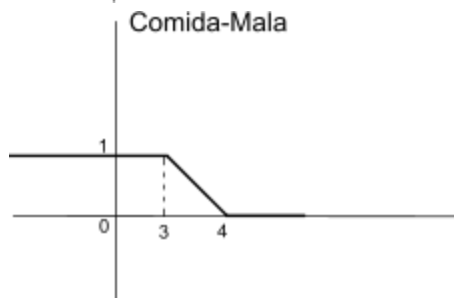
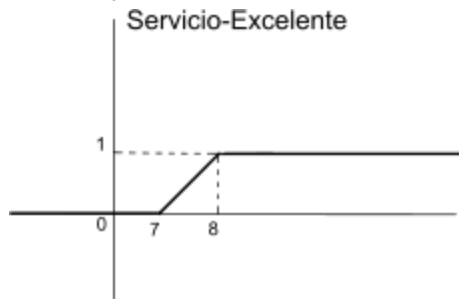
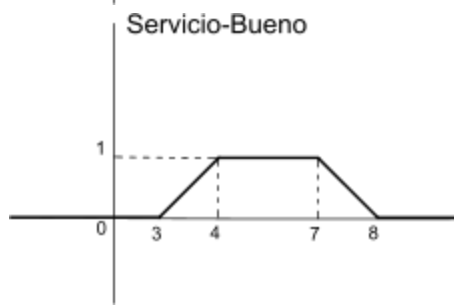
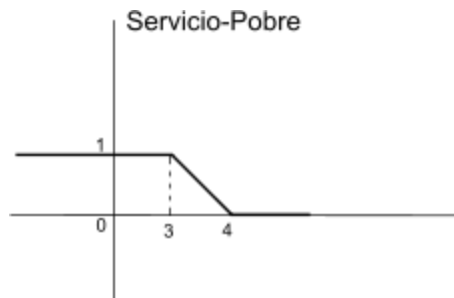
Las reglas son:

R1: servicio = pobre AND comida = mala => propina = poca

R2: servicio = buen AND comida = regular => propina = promedio

R3: servicio = excelente OR comida = deliciosa => propina = generosa.

Los gráficos para cada característica son:



Se emplearon 4 casos de prueba que devolvieron las siguientes salidas:

```
Ejemplo#2
Sevicio: 3.5, comida: 3 => propina: 17.5
Sevicio: 5, comida: 4 => propina: 20
Sevicio: 4, comida: 8 => propina: 50
Sevicio: 9, comida: 5 => propina: 50
```