

## Características del Sistema de Inferencia Propuesto

El sistema de inferencia que se propone cuenta con la implementación de tres funciones de membresía o pertenencia, la triangular, la trapezoidal y la gaussiana, dos métodos de agregación, Mamdani y Larsen, estos fueron seleccionados ya que son más factibles para problemas donde no se conocen un gran número de datos y se puede trabajar con un conjunto salida haciendo recorte o escalado. El sistema cuenta con todos los métodos de desdifusificación.

La implementación comienza con la clase **FuzzySet**, que es la que refleja un conjunto difuso y contiene las operaciones que se le pueden hacer.

Un **FuzzySet** se crea con un nombre, un valor inicial y un valor final con los que se crea un dominio de función discreto, cuenta con una función de membresía definida y así se calculan sus valores de imagen (ej. “Cerca”, de 0 hasta 8km con distribución trapezoidal). Las operaciones entre conjuntos como unión, escalado y recorte también aparecen en la definición, así como los métodos para desdifusificar. [1]

```
class FuzzySet:
    def __init__(self, name, min_val, max_val):
        self.name = name
        self.min_val = min_val
        self.max_val = max_val
        self.domain = list(range(min_val, max_val+1))
        self.image = []
```

```
def union(self, other_set): ...
```

```
def multiply(self, value): ...
```

```
def cutter(self, value): ...
```

```
def bisector(self): ...
```

```
def maxims(self): ...
```

```
def centroid(self): ...
```

**FuzzyVariable** es la clase que define a las variables que entran en el sistema, cada una contiene un número de **FuzzySet** que son los que determinan su comportamiento. Esta variable se crea con un nombre y los valores extremos que abarca, la clase contiene métodos para agregar cada tipo de **FuzzySet** y los métodos para difusificar un valor en esta variable, ya sea de dominio o de imagen, en el primer caso se calcula su valor de pertenencia y en el segundo se hace el recorte o escalado a partir del valor dado. Existe además un método adicional para graficar esta variable que muestra cómo se distribuyen los conjuntos que la forman. [2]

```
class FuzzyVariable():
    def __init__(self, name, min_val, max_val):
        self.name = name
        self._sets={}
        self.max_val = max_val
        self.min_val = min_val
```

Las reglas del sistema son tratadas en la clase *FuzzyRule*, a partir de una lista de antecedentes, el operador lógico que los relaciona y un consecuente. El antecedente y el consecuente son tuplas que contienen a la variable, el nombre del conjunto que interviene y un valor booleano opcional que toma False para el consecuente.

```
[(position, 'Far'), (quality, 'Bad')], (attendance, 'Low', False), 'or')
```

Regla 3: formada por los antecedentes, el operador y el consecuente

Las reglas se evalúan a partir de la entrada de valores y en dependencia del método de agregación que se emplee, yendo por cada antecedente, difusificando el conjunto con el valor de entrada, luego se aplica el operador lógico (se busca el mínimo para el *and* y el máximo para el *or*) entre los antecedentes, y el valor de imagen se pasa al consecuente para recortar o escalar. [3]

```
def evaluate_rule(self, input_values):
    eval_value = []

    for var, set_name in self.r_antecedents:
        for in_var, value in input_values:
            if in_var==var:
                eval_value.append(var.fuzzify(value, set_name))

    image_val = eval(f'{self.op}({eval_value})')

    return self.r_consequent[0].fuzzify(image_val, self.r_consequent[1], self.r_consequent[2])
```

Método para evaluar una regla del sistema

*FuzzySystem* es la clase que se encarga de realizar el proceso descrito anteriormente para cada regla del sistema (que son previamente agregadas) y luego de unir los conjuntos resultantes y aplicar el método de desdifusificación deseado.

## Propuesta de Problema a solucionar mediante inferencia difusa

El problema consiste en obtener un sistema capaz de predecir qué porcentaje de asistencia puede tener una sala de teatro, en dependencia de la calidad de la obra que va a colocar y de la distancia a la que se encuentra del centro de la ciudad (se toma esta como la zona más poblada). El sistema tiene relevancia para la preparación de eventos, catering, etc.

Se definen entonces tres tipos de variables, Posición, Calidad y Asistencia. La primera puede tomar tres valores principales: “Cerca” que sería hasta unos 8 km a la redonda desde el centro, “Media”, entre 6 y 15 km, y “Lejos”, desde los 14km hasta los 20, que sería el último valor a considerar (estamos situándonos en un municipio o ciudad pequeña). La calidad de la obra, se puede extraer de una crítica o reseña, y adecuar los valores de 0 a 10 en “Mala”, “Buena” y “Excelente”. Estas dos variables son de dominio discreto. Así mismo la asistencia (que es un porcentaje) se va a dividir en “Baja”, de 20 a 40%, “Moderada”, entre 30 y 70%, y “Alta”, que sería cuando la sala pase del 60% de ocupación.

El sistema tiene tres reglas principales que rigen su comportamiento:

- Regla 1:        *si Posición es “Cerca” o Calidad es “Excelente” > Asistencia es “Alta”*
- Regla 2:        *si Posición es “Media” o Calidad es “Buena” > Asistencia es “Moderada”*
- Regla 3:        *si Posición es “Lejos” o Calidad es “Mala” > Asistencia es “Baja”*

## Consideraciones obtenidas a partir de la solución del problema con el sistema de inferencia implementado

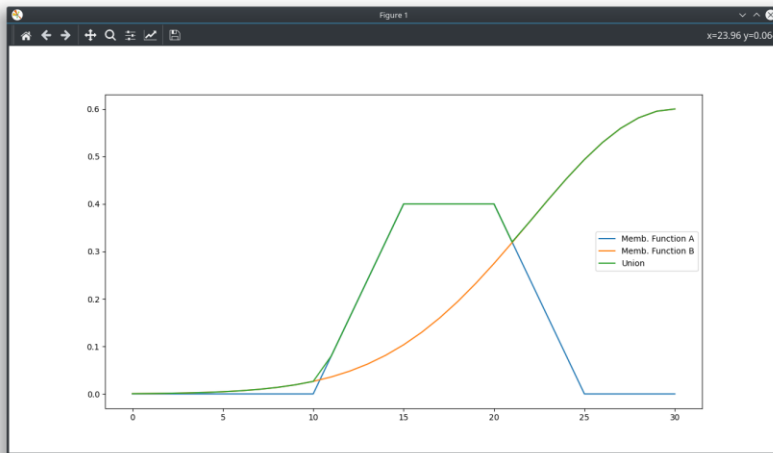
Al correr este sistema un número de veces no despreciable, se pudo determinar que hay varios factores que modifican su resultado, como lo son las distribuciones de los conjuntos difusos que tiene cada variable, la rigurosidad con que se redactan las reglas y sobre todo el método de desfusificación que se emplee para obtener el valor final, en este caso, el más acertado (más en acuerdo con el valor esperado) es el método del centroide, este es el más representativo, seguido de la bisectriz. Los métodos que involucran a los máximos, son de más fácil implementación, pero en ocasiones cuando el conjunto de salida brinda un único máximo ubicado en un extremo del conjunto, pueden resultar poco confiables y dar valores fuera de pronóstico.

La distribución trapezoidal también influye mucho ya que en su meseta para cada valor del dominio corresponde la misma imagen, lo que en algunos casos puede no ser el comportamiento deseado, así que para variables que se deseen tengan pequeñas variaciones visibles, no se recomienda.

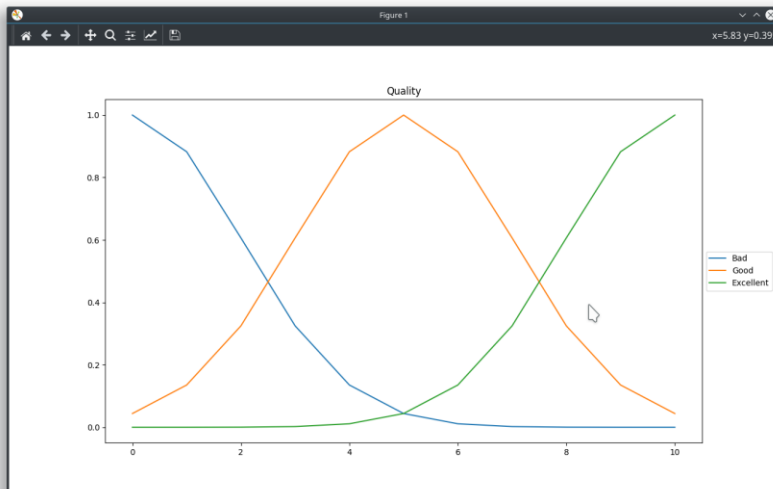
Las reglas, son un factor de mucho cuidado, ya que si no son exhaustivas pueden no llegar a cubrir todos los valores de entrada y dejar algunos sin obtener, las reglas pueden llegar a crecer exponencialmente con el número de conjuntos de cada variable, así que se recomienda, usar reglas con el operador “or”, como en el problema propuesto.

## Anexos

### 1. Unión entre conjuntos difusos



### 2. Variable de Calidad con distribución gaussiana para sus conjuntos



### 3. Recorte y escalado de un conjunto a partir de un valor

