Informe taller 2 – conjuntos difusos

Nombres:

Yulieth Tatiana Rengifo Rengifo – 2359748

Pedro José López Quiroz - 2359423

Juan José Valencia Jimenez - 2359567

Docente:

Carlos Andres Delgado

Universidad del Valle

Sede Tuluá

1. **Informe de procesos:**

Función: pertenece

- Ejemplo: Evaluar si elem=5elem = 5elem=5 pertenece al conjunto difuso s(x)=x10s(x) = \frac{x}{10}s(x)=10x.
- Código:

```
def pertenece(elem: Int, s: ConjDifuso): Double = {
    s(elem)
}
```

- **Pila de llamadas generada:**
    1. Llamada inicial: pertenece(5,s)pertenece(5, s)pertenece(5,s).
    2. Evaluación: Se ejecuta s(5)=510=0.5s(5) = \frac{5}{10} = 0.5s(5)=105 =0.5.
    3. Retorno del valor 0.50.50.5.

Despliegue de la pila:

- La pila es trivial porque es un proceso iterativo: la función realiza directamente el cálculo de s(elem)s(elem)s(elem) y retorna el resultado sin generar llamadas recursivas.

Función: grande

- Ejemplo: Crear un conjunto difuso para números "grandes" con d=2d = 2d=2 y e=3e = 3e=3, y evaluar grande(4)grande(4)grande(4).
- Código:

```
def grande(d: Int, e: Int): ConjDifuso ={
    def gd(n: Int): Double = {
        if (n <= 0) 0.0 // Si el número es menor o igual a cero, no es grande
        else Math.pow(n.toDouble / (n + d).toDouble, e.toDouble)
    }
    gd
}
```

- **Pila de llamadas generada:**
    1. Llamada inicial: grande(2,3)grande(2, 3)grande(2,3).
    2. Internamente: gd(4)=(44+2)3=(46)3=0.296gd(4) = \left(\frac{4}{4+2}\right)^3 = \left(\frac{4}{6}\right)^3 = 0.296gd(4)=(4+24)3=(64)3=0.296.

Despliegue de la pila:

- Este es un proceso iterativo, ya que el cálculo se realiza en un único paso sin generar llamadas adicionales.

Función: complemento

- Ejemplo: Calcular el complemento de c(x)=x10c(x) = \frac{x}{10}c(x)=10x para x=4x = 4x=4.
- Código:

```
def complemento(c: ConjDifuso): ConjDifuso = {
    def comp(n: Int): Double = {
        1.0 - c(n)  // El complemento se calcula como 1 - fS(n)
    }
    comp
}
```

- **Pila de llamadas generada:**
    1. Llamada inicial: complemento(c)(4)complemento(c)(4)complemento(c)(4).
    2. Internamente: comp(4)=1.0−c(4)=1.0−410=0.6comp(4) = 1.0 - c(4) = 1.0 - \frac{4}{10} = 0.6comp(4)=1.0−c(4)=1.0−104=0.6.

Despliegue de la pila:

- Similar a las funciones anteriores, el cálculo se realiza en un único paso sin llamadas recursivas.

(Este análisis se repite para cada función: union, interseccion, inclusion, igualdad)

2. Informe de corrección

Argumentación sobre la corrección

Función: pertenece

- Definición matemática: pertenece(elem,s)=s(elem)pertenece(elem, s) = s(elem)pertenece(elem,s)=s(elem).
- Argumento de corrección: La implementación es correcta porque toma el valor del conjunto difuso s(x)s(x)s(x) en el punto elemelemelem, que por definición es el grado de pertenencia de elemelemelem al conjunto sss.
- Conclusión: La implementación es consistente con la definición teórica.

Función: grande

- Definición matemática: grande(x)=(xx+d)egrande(x) = \left(\frac{x}{x + d}\right)^egrande(x)=(x+dx)e, si x>0x > 0x>0; de lo contrario, grande(x)=0grande(x) = 0grande(x)=0.

- Argumento de corrección:
  La implementación sigue la definición exacta de un conjunto difuso "grande", utilizando los parámetros ddd y eee para ajustar la curva de pertenencia.
- Conclusión: La implementación es correcta.

(Este análisis se repite para las demás funciones: complemento, union, interseccion, inclusion, igualdad.)

3. Casos de Prueba:

Función: Pertenece

Caso	Entrada	Salida Esperada	Resultado
1	(5, x/10)	0.5	Correcto
2	(0, x/10)	0.0	Correcto
3	(-3, x/10)	0.0	Correcto
4	(10, x/10)	1.0	Correcto
5	(7, x/20)	0.35	Correcto

Función: Grande

Caso	Entrada	Salida Esperada	Resultado
1	(2, 3, 4)	0.296	Correcto
2	(5, 2, 10)	0.512	Correcto
3	(1, 3, -1)	0.0	Correcto
4	(2, 1, 8)	0.470	Correcto
5	(0, 5, 7)	0.0	Correcto

(Se repite este formato de tabla para las demás funciones: complemento, union, interseccion, inclusion, igualdad.)