Universidad del Valle - Sede Tulua

Fundamentos de Programación Funcional y Concurrente

Taller 2: Funciones de Alto Orden

Juan Manuel Perez Cruz - 2266033
Juan Jose Millán - ?
Daniel Ceballos - ?
Juan Diego Escobar T - 2359393

La representación de los conjuntos difusos como funciones facilita el uso de funciones de alto orden para definir operaciones sobre estos conjuntos. Cada operación, como la unión, intersección y complemento, se implementa como una función que recibe conjuntos (representados como funciones) y devuelve un nuevo conjunto.

Ejemplo de Implementación: Unión de Conjuntos Difusos, la función Union recibe dos conjuntos difusos (cd1 y cd2) y devuelve un conjunto que representa la unión. Para cada valor x, Union calcula el máximo grado de pertenencia entre ambos conjuntos, según la fórmula:

Informe de Procesos:

Para implementar los conjuntos difusos y sus operaciones, se utilizaron funciones de alto orden que permiten trabajar con conjuntos como funciones inmutables. Cada operación recibe conjuntos y devuelve nuevos conjuntos basados en el cálculo requerido para cada caso.

Las operaciones incluyen:

- **Pertenencia:** Permite evaluar el grado de pertenencia de un elemento a un conjunto difuso específico.
- **Complemento:** Define el conjunto complementario, donde el grado de pertenencia de cada elemento es invertido a través de la fórmula 1-f(x)1 - f(x)1-f(x).
- **Unión e Intersección:** Calculan los conjuntos resultantes al unir o intersectar dos conjuntos difusos, aplicando el máximo y mínimo grado de pertenencia, respectivamente.
- **Inclusión:** Verifica si todos los elementos de un conjunto tienen un grado de pertenencia menor o igual en otro conjunto.
- **Igualdad:** Compara dos conjuntos para determinar si tienen exactamente el mismo grado de pertenencia para cada elemento en un rango específico.

Cada operación se maneja mediante funciones de alto orden, manteniendo la representación funcional de los conjuntos y facilitando su aplicación modular.

También se resalta que la función inclusión es un ejemplo clave del uso de recursión para evaluar si un conjunto difuso está incluido en otro. Esto se realiza evaluando cada elemento en un rango específico mediante una función auxiliar que verifica que el grado de pertenencia en el primer conjunto no exceda el del segundo conjunto. Este proceso utiliza recursión de cola para gestionar el recorrido de elementos sin necesidad de estructuras adicionales.

Informe de Corrección:

Argumentación de la Corrección usando Notación Matemática
Aquí explicamos cómo cada función cumple con lo que se espera en términos de conjuntos difusos y mostramos por qué el código es correcto.

Pertenece

La función Pertenece recibe un conjunto difuso s y un elemento, devolviendo el valor $f_s(x)$, que representa el grado de pertenencia de x en el conjunto s. Matemáticamente, esta función verifica que el valor esté entre 0 y 1, como debe ser en un conjunto difuso. Cada llamada a Pertenece devuelve el grado de pertenencia exacto para un x específico, sin alterar el conjunto.

grande(d: Int, e: Int): ConjuntoDifuso

Esta función crea un conjunto difuso que representa números grandes. La fórmula usada es: $f(x) = (\frac{x}{x+d})^e$

donde d y e son parámetros que ajustan la sensibilidad. Para valores altos de x, $\frac{x}{x+d}$ se acerca a 1, lo que da como resultado que $f(x)$ también se aproxime a 1, cumpliendo así la idea de que x es grande. Por cada llamada, la función calcula este valor, siempre manteniendo el resultado entre 0 y 1, como debe ser para un conjunto difuso.

Complemento

La función complemento devuelve el valor complementario de pertenencia de un elemento usando:

$f\neg s(x) = 1 - f_s(x)$ Esta fórmula invierte el grado de pertenencia sin salirse del rango $[0, 1]$. Entonces, si en s(x) el valor es alto, en el complemento será bajo, y viceversa. Cada vez que se llama a complemento, se aplica esta fórmula directamente sobre el valor original de s(x), dándonos el complemento esperado.

Unión e Intersección

Estas operaciones se definen así:

Unión:

Calcula el valor máximo de pertenencia entre dos conjuntos cd1 y cd2 para cada elemento:

$f_{s1} \cup_{s2}(x) = max(f_{s1}(x), f_{s2}(x))$

Esto asegura que, para cada elemento, el resultado tiene el mayor valor de pertenencia que haya en cd1 o cd2, sin salirse del rango $[0, 1]$.

Intersección:

Calcula el mínimo grado de pertenencia para cada elemento:

$f_{s1} \cap_{s2}(x) = min(f_{s1}(x), f_{s2}(x))$

Aquí el valor de pertenencia es el más bajo de los dos conjuntos, asegurando que el resultado siga siendo válido en el contexto de conjuntos difusos.

Inclusión e Igualdad

Inclusión:

Esta función evalúa si todos los elementos de cd1 tienen un grado de pertenencia menor o igual al de cd2:

$f_{S1}(x) \leq f_{S2}(x), \forall x \in [0, 1000]$

La función compara cada valor en el rango $[0, 1000]$. Si en algún punto cd1(x) es mayor que cd2(x), devuelve false; si no, devuelve true, confirmando que cd1 está incluido en cd2.

Igualdad:

Esta función verifica si cd1 y cd2 son iguales, es decir, si tienen el mismo grado de pertenencia para cada elemento. Matemáticamente, esto se expresa como:

$S1 = S2 \Leftrightarrow f_{S1}(x) = f_{S2}(x)\ \forall x \in [0,1000]$

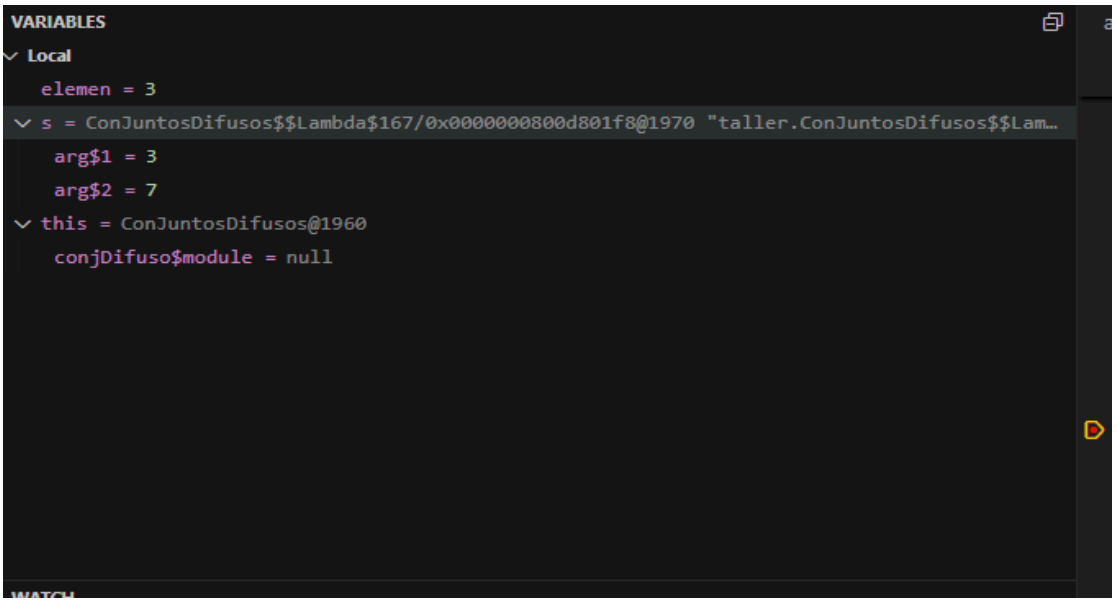
La función utiliza inclusión dos veces para verificar la inclusión mutua, lo cual significa que cd1 debe estar incluido en cd2 y viceversa. Si ambas condiciones se cumplen, los conjuntos son iguales.

Casos de prueba:

Conjuntos difusos:

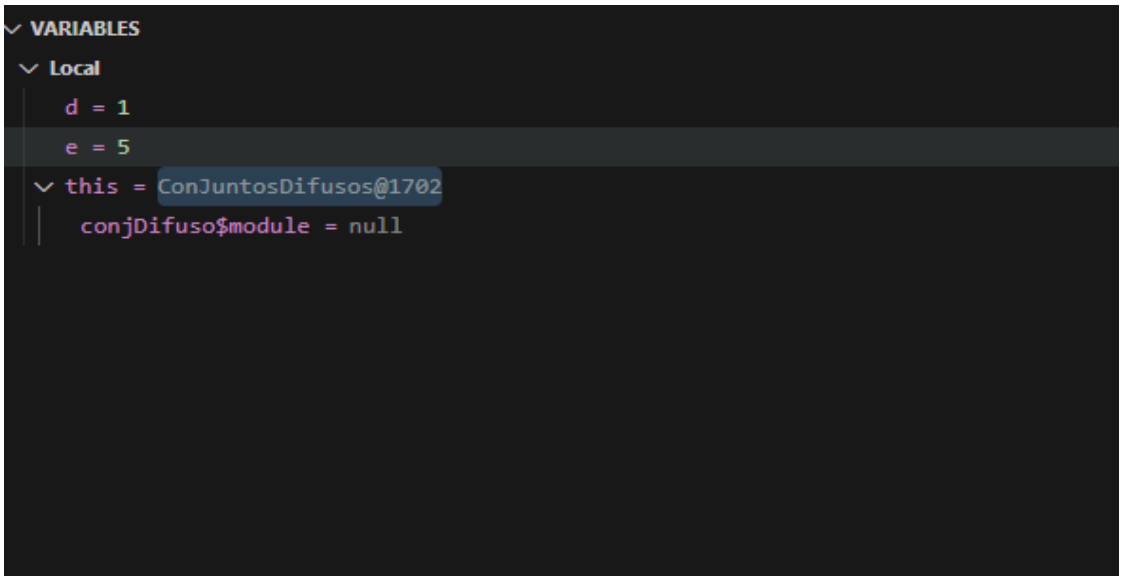
test "Pertenece 3 en muchoMayorQue(3, 7)"

La prueba test ("Pertenece 3 en muchoMayorQue(3, 7)") evalúa cómo funciona la función Pertenece al determinar el grado de pertenencia del número 3 dentro del conjunto muchoMayorQue(3, 7). En este contexto, como el valor 3 es igual al parámetro a en la definición de muchoMayorQue(3, 7), de acuerdo con la lógica de mma, su grado de pertenencia debería ser 0.0.



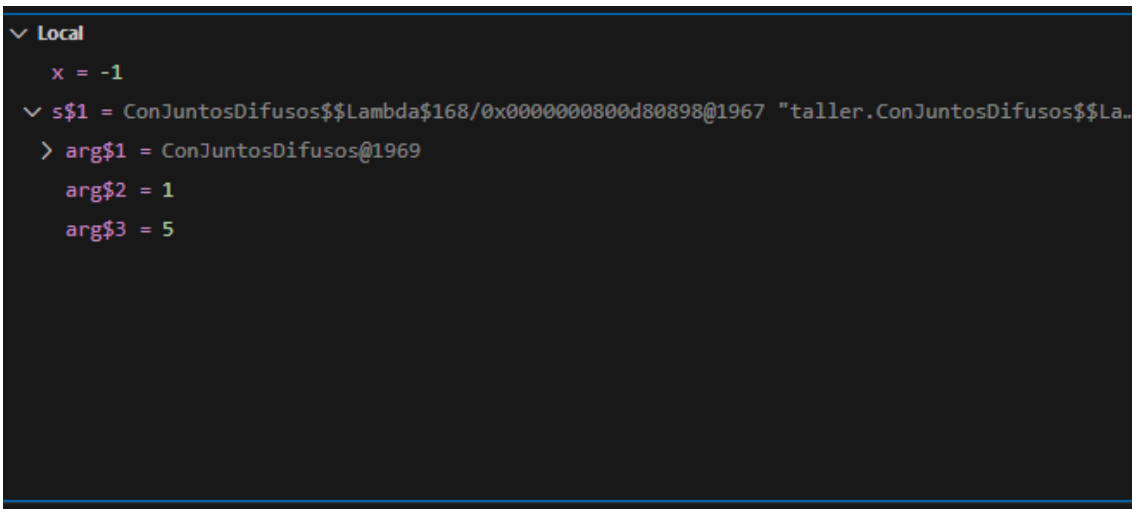
test "Grande con valor normal 1"

La prueba test("Grande con valor normal 1") verifica que el grado de pertenencia obtenido al evaluar grande(1, 5)(3) sea exactamente 0.237. Esta expresión, grande(1, 5)(3), indica que el test está calculando el grado de pertenencia del valor 3 en el conjunto difuso definido por grande con los parámetros d = 1 y e = 5.



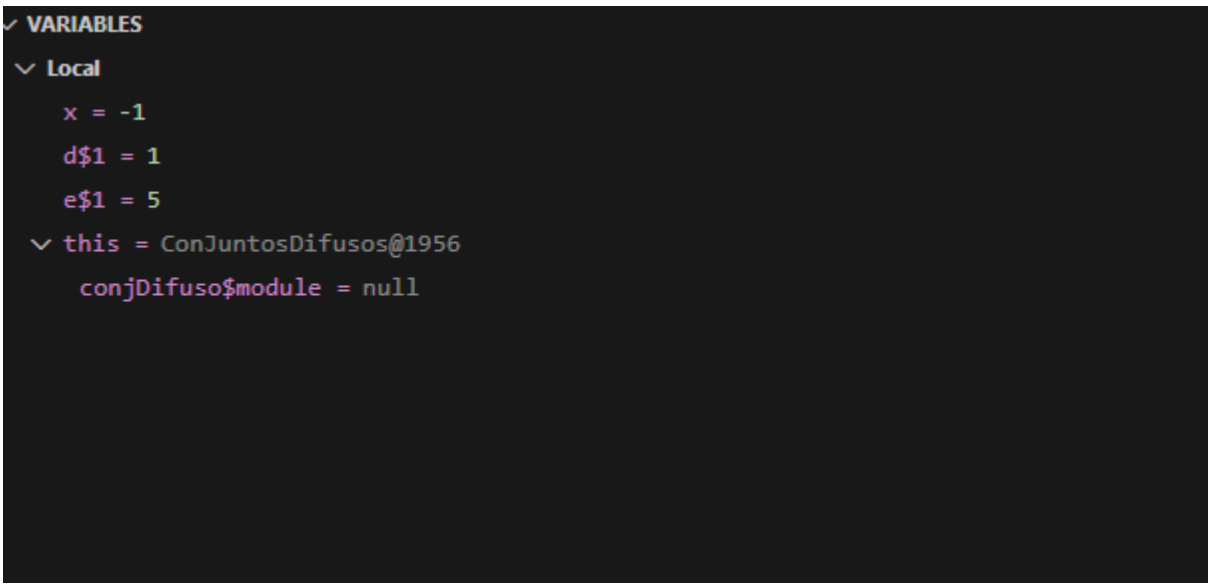
test "Complemento con valor fuera de rango"

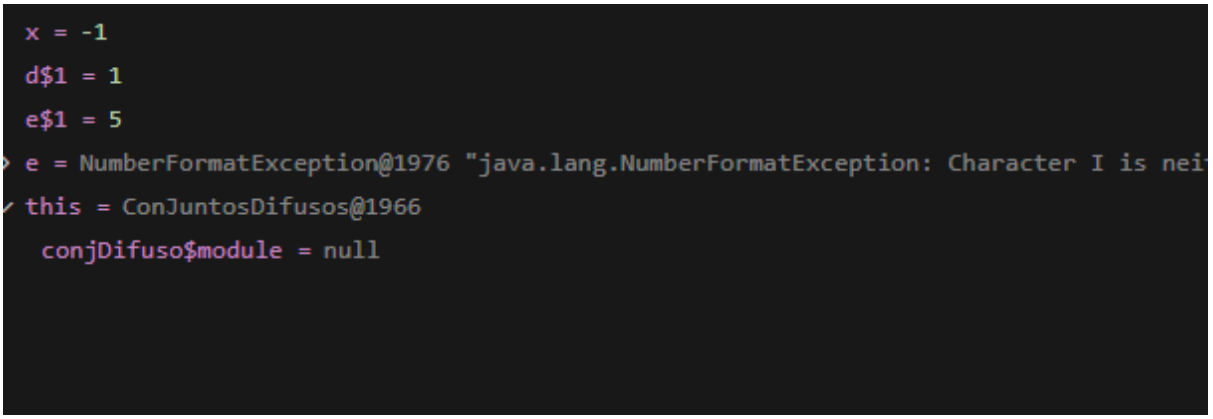
El test test("Complemento con valor fuera de rango") verifica que al intentar calcular el complemento de un valor fuera del rango esperado (específicamente -1 en este caso), la función lance una excepción.



test "Union con valor fuera de rango"

verifica que al intentar calcular la unión de dos conjuntos difusos (grande(1, 5) y muchoMayorQue(3, 7)) para un valor fuera del rango esperado (-1 en este caso), se lance una excepción.




x = -1
d$1 = 1
e$1 = 5
e = NumberFormatException@1976 "java.lang.NumberFormatException: Character I is nei
this = ConJuntosDifusos@1966
conjDifuso$module = null