

## Técnicas, Entornos y Aplicaciones de Inteligencia Artificial

### Evaluación Práctica-1: Fuzzy-CLIPS. 2020-21.

Nombre: \_\_\_\_\_

*Al realizar esta prueba de evaluación ACEPTO la "cláusula de veracidad" por la que no recibiré ni daré ayuda en esta prueba y garantizo la autoría del 100% de los resultados.*

*El incumplimiento por mi parte de los deberes derivados de las buenas prácticas de honestidad académica podrá dar lugar a la adopción de las medidas contenidas en la Normativa de Integridad Académica del alumnado de la UPV.*

- 1) Subid a Poliformat el fichero .clp correspondiente a la práctica ya realizada, incluyendo extensiones.
- 2) Contestad a las preguntas siguientes, rellenando los huecos con las respuestas (doc, txt). Se asume que se parte de la práctica ya realizada, incluyendo extensiones.
- 3) Subid un fichero con el código modificado resultante tras realizar todo el examen.

Tiempo: 1 hora.

**PREVIAMENTE:** Confirmad que usáis los valores difusos dados en el boletín de las prácticas. La fusificación de valores CRISP debe hacerse con un  $\delta=0$ , es decir: (`fuzzify Var_difusa Valor-CRISP 0`)

1. (2 puntos, Tiempo estimado:10') Utilizando el código desarrollado en la práctica, obtened "el valor VUE (con un valor CRISP)", que resulta a partir de los valores en cada uno de los siguientes casos. Aplicad la defusificación por `moment-defuzzify` y por `maximum-defuzzify` (no es necesario poner decimales).

	Caso-1	Caso-2	Caso-3
Categoría (puntos):	45	70	110
Edad Aparente (años):	40	5	10
Nº de Ventanas:	2	4	7
Valor VUE crisp (momentum)	1192	3500	5094
Valor VUE crisp (maximum)	1500	3500	5500

**Responde a la siguiente pregunta:** ¿Es posible asertar de forma sucesiva varios y distintos valores difusos a una misma variable difusa?.

Por ejemplo (`assert (edad-aparente Nuevo)`), (`assert (edad-aparente Viejo)`), (`assert (edad-aparente very Viejo)`).  
Justifica la respuesta.

**Sí es posible. No hay problema en asertar diferentes valores difusos a una misma variable difusa. En este caso concreto resultaría como:** (`assert (edad-aparente OR [Nuevo Viejo very Viejo] )`).

**2. (3 puntos, Tiempo estimado: 15')** Extender el código realizado para **incorporar nuevo conocimiento**:

Además de los valores-fuzzy (Reciente, Nuevo, Medio, Viejo) ya introducidos sobre la variable 'Edad-Aparente', se ha decidido introducir dos nuevos valores:

**Antigua, con la función de pertenencia (s 50 80).**

**Rehabilitada, con la función de pertenencia (pi 20 50).**

Además, se añaden **dos nuevas reglas** a las ya introducidas en la práctica:

	Categoría-Vivienda	Edad_Aparente	⇒	VUE-DIF
10	Económica o Estándar	very Antigua	⇒	Bajísimo
11	Intermedia o Alta	Rehabilitada	⇒	No Bajo

Indicad la modificación necesaria para incluir los nuevos valores difusos de la variable 'Categoría-Vivienda' y las dos nuevas reglas:

Modificación de la variable 'Edad-Aparente':

```
(deftemplate edad ;Variable difusa
  0 100 años ;Universe
  ((reciente (0 1) (12 0))
   (nuevo (0 0) (12 1) (24 0))
   (medio(24 0) (36 1) (48 0))
   (viejo (48 0) (60 1))
   (antigua (s 50 80))
   (rehabilitada (pi 20 50))))
```

Nueva Regla-1

```
(defrule r10
  (categoria-dif economica OR estandar)
  (edad very antigua)
  =>
  (assert (vuedif bajisimo)))
```

Nueva Regla-2

```
(defrule r11
  (categoria-dif intermedia OR alta)
  (edad rehabilitada)
  =>
  (assert (vuedif not [ bajo ])))
```

Obtener los nuevos valores VUE (crisp) resultantes de estas modificaciones, en estos casos:

	Caso-1	Caso-2
Categoría (puntos):	55	115
Edad Aparente (años):	24	40
Nº de Ventanas:	6	3
Valor VUE crisp (momentum)	3151	5100
Valor VUE crisp (maximum)	5500	3250

3. (1 puntos, Tiempo estimado: 10') Se desea añadir una nueva información sobre las viviendas relativa a la puntuación sobre los servicios de transporte público cercanos, que se puntuarán de 0 a 10. El valor de cada vivienda se debe añadir como un **nuevo atributo (slot) 'ServiciosTransporte'**. Indicad la modificación necesaria en la representación realizada.

```
(deftemplate casa ;Clase Casa
  (slot nombre (type SYMBOL))
  (slot categoria (type INTEGER))
  (slot edad-aparente (type INTEGER))
  (slot ventanas (type INTEGER))
  (slot serviciostransporte (type INTEGER)) ; Atributo añadido
  (slot vue-max (type FLOAT))
  (slot vue-momentum (type FLOAT)))
```

Contestad razonablemente: Si no se añade ninguna regla más, ¿Funcionará el sistema exactamente igual que antes, o hay que modificar algo (indícalo en su caso)?

**No hay que modificar nada. La inclusión del nuevo slot no afecta a las reglas planteadas**

4. (4 puntos, Tiempo estimado: 15')

Una vez evaluada una vivienda, se desea obtener una relación de las viviendas con mayores o menores prestaciones. Concretamente, se desea que el sistema emita un mensaje por consola del tipo:

- "Vivienda de alto-standing", si el valor VUE difuso obtenido para una vivienda es 'extremely' alto y tiene un valor del atributo 'ServiciosTransporte' mayor que 7.
- 'Vivienda de bajas prestaciones', si tiene valor VUE crisp (por máximo) menor que 3.000, tiene menos de 3 ventanas y un valor del atributo 'ServiciosTransporte' menor que 3.

Indicad la modificación de código necesaria, con la(s) regla(s) correspondiente(s).

**Es importante la prioridad adecuada a cada regla y el acceso a la información de los slots de la vivienda.**

```
(defrule alto-standing
  (declare (salience -10)) ; Una vez obtenido el valor difuso final para vuedif
  (casa (nombre ?nom) (serviciostransporte ?s) )
  (vuedif extremely alto)
  (test (> ?s 7))
  => (printout t " Vivienda de alto-standing " ?nom crlf))

(defrule bajas-prestaciones ;
  (declare (salience -300)) ; Una vez obtenido el valor crisp de vuedif
  (casa (nombre ?nom) (ventanas ?v) (serviciostransporte ?s) (vue-max ?vue))
  (test (< ?vue 3000))
  (test (< ?s 3)) (test (< ?v 3))
  => (printout t "Vivienda de bajas prestaciones" crlf))
```