



Técnicas, Entornos y Aplicaciones de Inteligencia Artificial

Videos Recomendados para su Visualización previa a la Clase (Tema-2)

DOCENCIA VIRTUAL

Finalidad:
Prestación del servicio Público de educación superior (art. 1 LOU)

Responsable:
Universitat Politècnica de València.

Derechos de acceso, rectificación, supresión, portabilidad, limitación u oposición al tratamiento conforme a políticas de privacidad:
<http://www.upv.es/contenidos/DPD/>

Propiedad intelectual:
Uso exclusivo en el entorno de aula virtual.
Queda prohibida la difusión, distribución o divulgación de la grabación de las clases y particularmente su compartición en redes sociales o servicios dedicados a compartir apuntes.
La infracción de esta prohibición puede generar responsabilidad disciplinaria, administrativa o civil



EJEMPLOS



Tema 2: Imprecisión e Incertidumbre

2.1: Razonamiento Impreciso. Lógica Difusa.

- Imprecisión de la Información. Lógica Difusa. Conjuntos Difusos. Operaciones.
- Proceso Inferencial: Fusificación, Defusificación, Inferencia Difusa.
- Entornos y Aplicaciones

2.2: Incertidumbre. Razonamiento Probabilístico.

- Conceptos básicos. Aplicación Teoría de la Probabilidad. Modelos simples.
- Redes Bayesianas

Bibliografía

- **Inteligencia Artificial: Un enfoque moderno.** Rusell, Norvig. Prentice Hall, 2004. Cap. 13, 14, 16
- **Inteligencia Artificial. Técnicas, métodos y aplicaciones.** Varios autores. McGraw Hill (2008) Cap.6-7
- **Inteligencia Artificial. Una nueva síntesis.** N. Nilsson McGraw Hill (2000).

Otras referencias:

- **FuzzyClips** (desarrollado por National Research Council , Canada)
- European Centre for Soft Computing (<http://www.softcomputing.es/>)

Imprecisión: Grado de precisión del conocimiento

Datos conocidos aproximadamente, Precisión de las medidas, Datos cualitativos y/o simbólicos, etc.

- **Hechos:** Hoy llueve '**mucho**', Hace '**mucho frío**', Es '**bastante cierto**' que...,
El síntoma duró **aprox 2 horas**, ...
- **Reglas:** Los hombres **ricos** son **felices**,
Si está **muy nuboso** entonces **probablemente** llueva **mucho**,
Los coches **caros** duran **mucho tiempo**
Si el agua está **muy fría**, abre **mucho** el grifo de la caliente

⇒ *Lógica difusa de Zadeh*

Definición de conjuntos (conceptos, predicados) difusos y extensión de las reglas de inferencia.

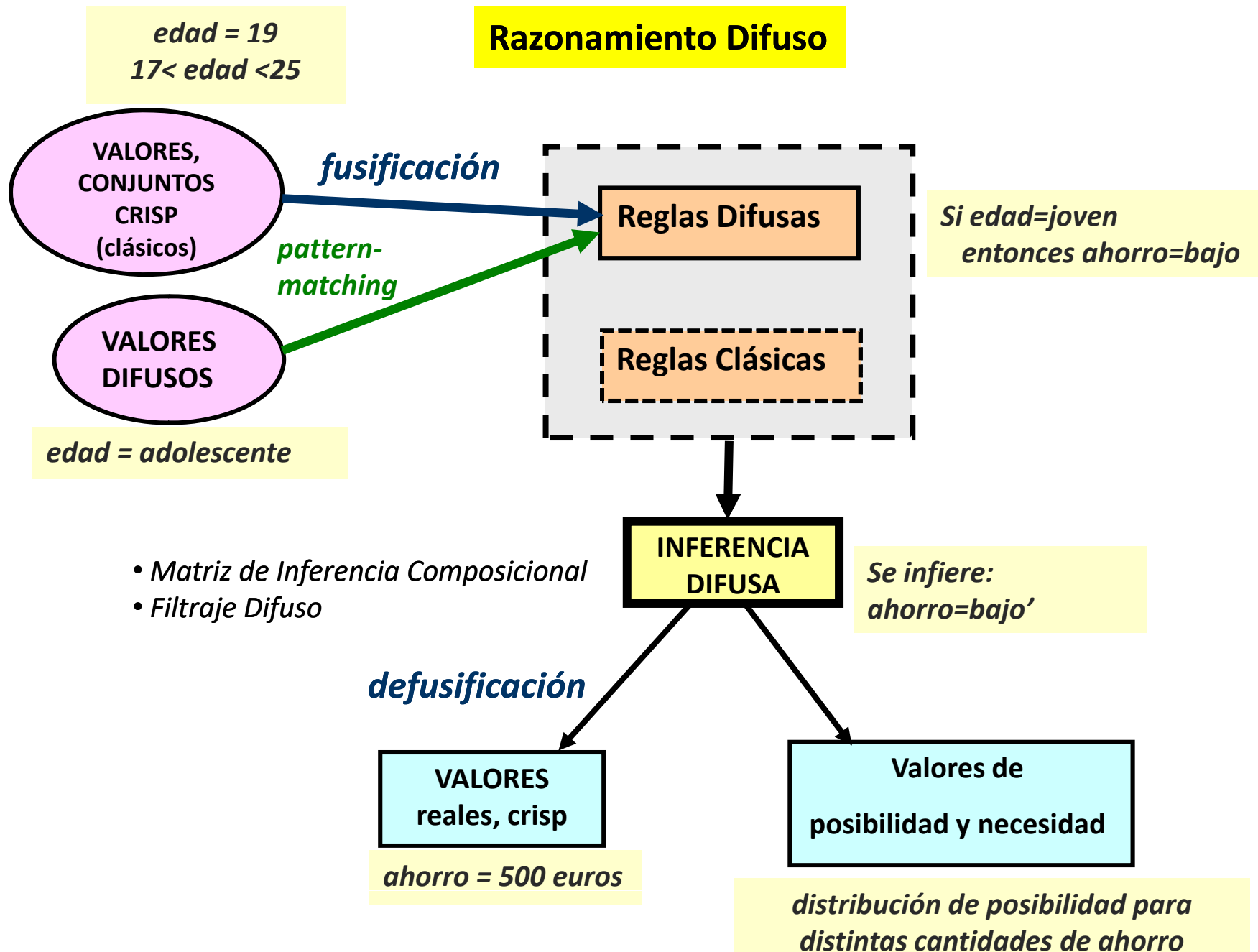
⇒ *Razonamiento difuso*

*El dolor empezó **hacia la 25ª** semana de gestación y duró **casi dos días**.*

***Casi a la vez** la paciente empezó a sufrir mareos y **fiebre alta**.*

***Unos dos meses** después apareció hipertensión arterial **severa**, y en la consulta de la **36ª** semana se detectó un **muy alto índice** de proteinuria.*

*Todo ello permite pensar que **unos días antes** se incrementó **considerablemente** el nivel de glucosa en sangre.*



Winners of the fuzzy logic video competition held by the IEEE Computational Intelligence Society



[An Egg-Boiling Fuzzy Logic Robot](#)

KIOS Research Center for Intelligent Systems and Networks, University of Cyprus

https://www.youtube.com/watch?v=J_Q5X0nTmrA

Otros:

➤ [An Introduction to Fuzzy Logic](#)

- [Fuzzy Logic: An Introduction](#)
DeMontfort University, Leicester, England.
- [Benjamin Franklin Medal Winner: Lotfi Zadeh](#)



Zojirushi Micom Fuzzy-Logic - Arrocera y calentador 10 cup blanco

Marca: ZOJIRUSHI

★★★★★ 208 valoraciones

No disponible.

Marca	Zojirushi
Color	Blanco
Material	Acero inoxidable
Capacidad	10 Tazas
Dimensiones del producto:	36.5 x 27.9 x 25.1 centímetros
largo x ancho x alto	

Acerca de este producto



Aspirador Escoba Ultimate Digital Fuzzy -Taurus

Referencia - 948900000

Etiquetas: aspirador escoba, Ultimate Digital Fuzzy

219 €

- 1 +

Añadir al carrito

Aspirador escoba Ultimate Digital Fuzzy, Driver, que lo convierte en un aspirador e: compacto, más ligero y más silencioso. S Brush Fuzzy es ideal para parquet y suelo

Cuenta con 80.000 rpm que garantizan l 2 kg de peso que asegura la máxima vers

Lavarropas Automático Drean 5 KG CONCEPT FUZZY LOGIC TECH V1 Blanco

★★★★★ 3 opiniones

[Descripción](#) [Especificaciones técnicas](#)



Lógica Difusa. Conjuntos difusos (Zadeh, 1979).



Conjunto Clásico P, definido en un Universo U, mediante un predicado $P(x): U \rightarrow \{0, 1\}$

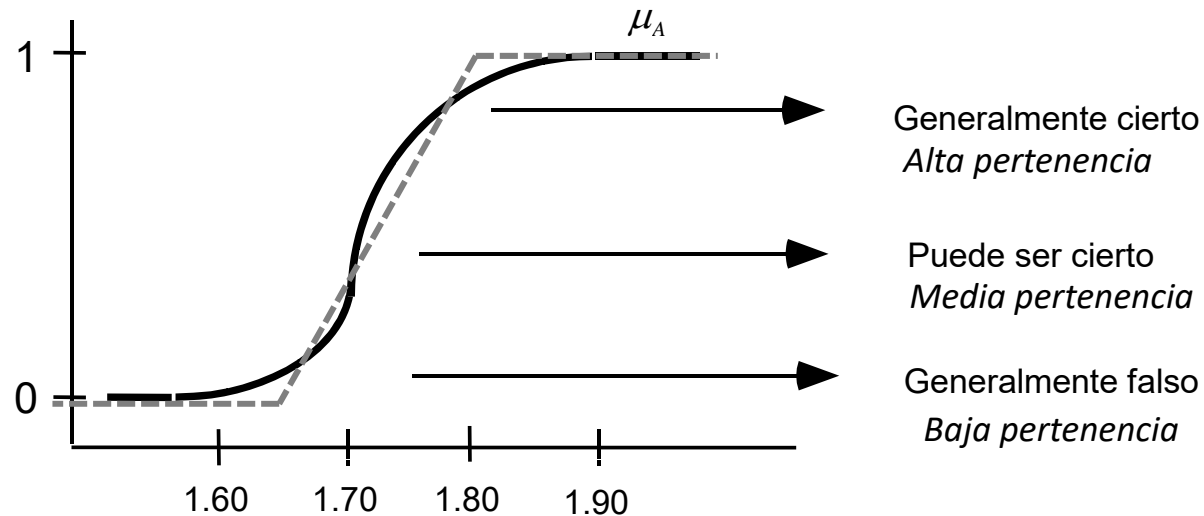
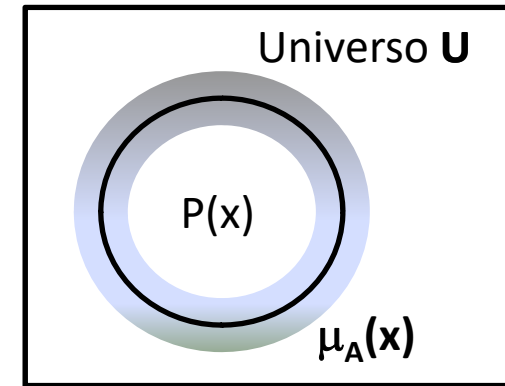
Conjunto difuso A, definido en un Universo U mediante una

Función de Pertenencia $\mu_A: U \rightarrow [0, 1]$

Ejemplo:

U: altura de personas (metros)

A: conjunto difuso “personas altas”



Variables y Valores Difusos

Variable difusa (o *variable lingüística*): *Estatura*

Valores de la variable difusa (o *valores lingüísticos*): *{alta, media, baja}*

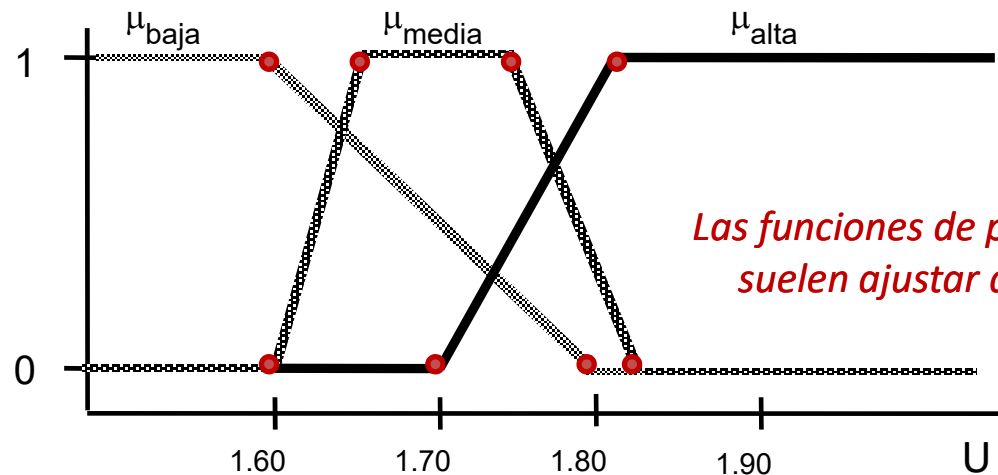
Conjuntos Difusos {Estatura-alta, Estatura-media, Estatura-baja} \Rightarrow Funciones de pertenencia

A1: conjunto difuso que representa estatura baja (μ_{baja})

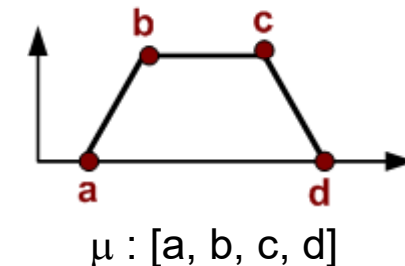
A2: conjunto difuso que representa estatura media (μ_{media})

A3: conjunto difuso que representa estatura alta (μ_{alta})

La 'pertenencia' valora la posibilidad, verosimilitud, creencia, preferencia, etc., de pertenecer a un conjunto.



Las funciones de pertenencia se suelen ajustar a trapezios.



La partición del dominio U (en f conjuntos difusos) puede tener como propiedades:

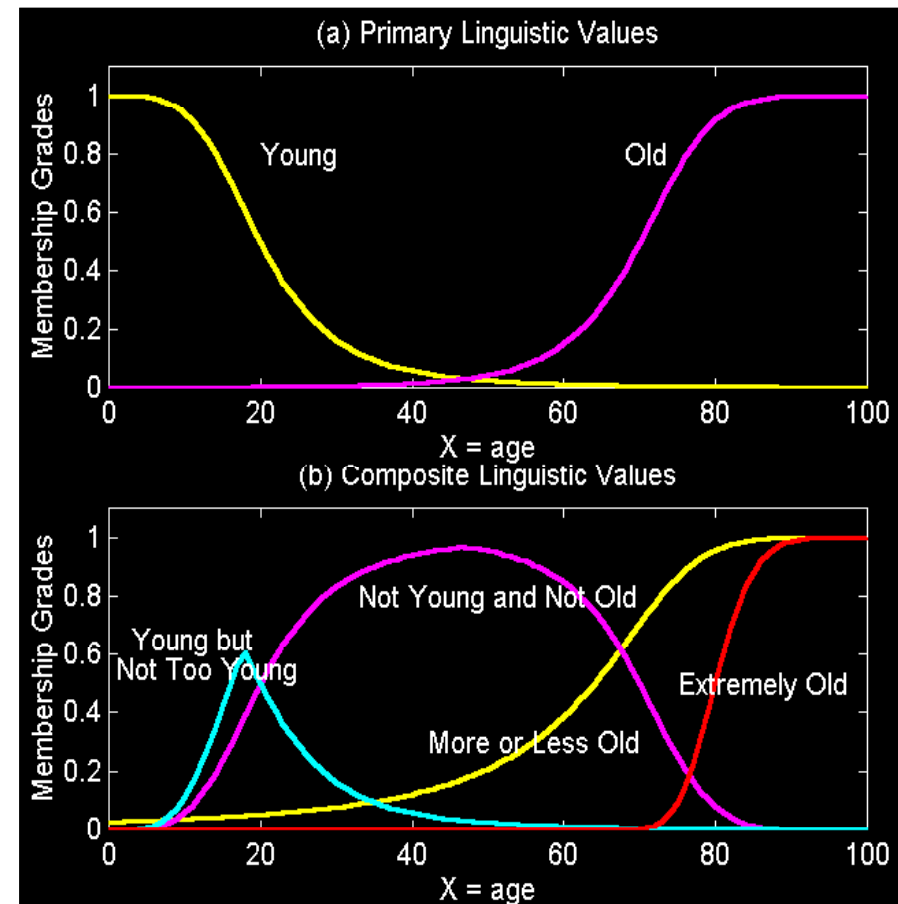
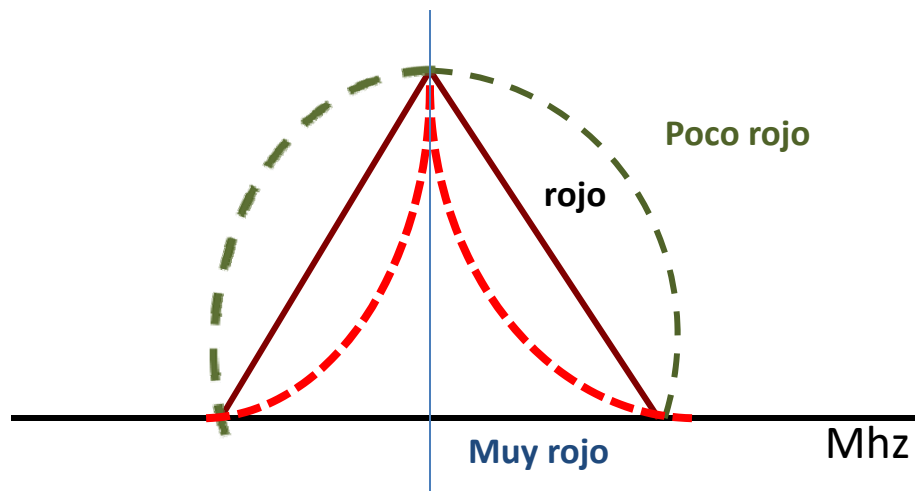
- **Completitud:** ningún valor del dominio queda fuera de la partición fuzzy.
- **Partición fuerte:** Suma de las funciones de pertenencia para cada valor del dominio es 1:

$$\sum_{\forall x \in U}^f \mu_f(x) = 1$$

Modificadores Lingüísticos:

Un valor difuso puede ser modificado mediante modificadores lingüísticos.
Existen estándares para la definición de modificadores lingüísticos.

X es A	$\Rightarrow \mu_A(x)$
X es extremadamente A	$\Rightarrow (\mu_A(x))^3$
X es muy A	$\Rightarrow (\mu_A(x))^2$
X es más-o-menos A	$\Rightarrow \sqrt{\mu_A(x)}$
X es algo A	$\Rightarrow (\mu_A(x))^{1/3}$
X no es A	$\Rightarrow 1 - \mu_A(x)$



Operaciones con conjuntos difusos

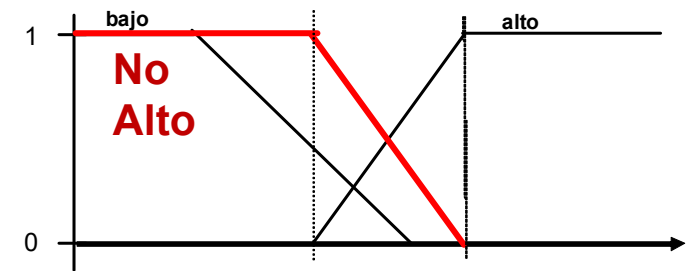
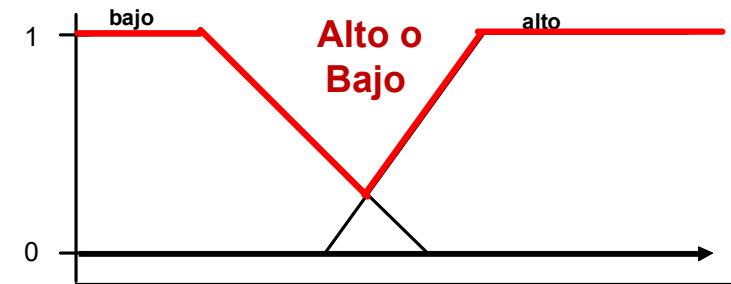
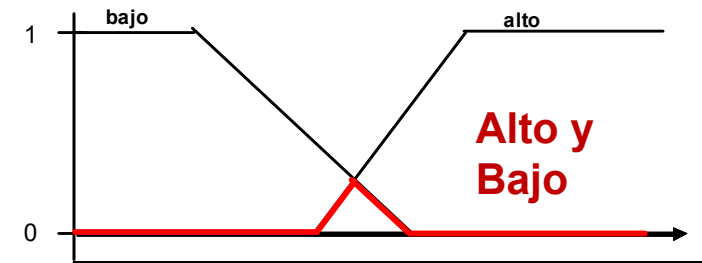
Igualdad de conjuntos ($A=B$): $\forall x \in U, \mu_A(x) = \mu_B(x)$

Inclusión de conjuntos $A \subseteq B$: $\forall x \in U, \mu_A(x) \leq \mu_B(x)$

Intersección de dos conjuntos $A \cap B$: $\min(\mu_A(x), \mu_B(x))$

Unión de dos conjuntos $A \cup B$: $\max(\mu_A(x), \mu_B(x))$

Complemento de un conjunto difuso A : $\mu_A^{-1}(x) = 1 - \mu_A(x)$



Ejemplos de información difusa

Tamaño: {pequeño, medio, grande}

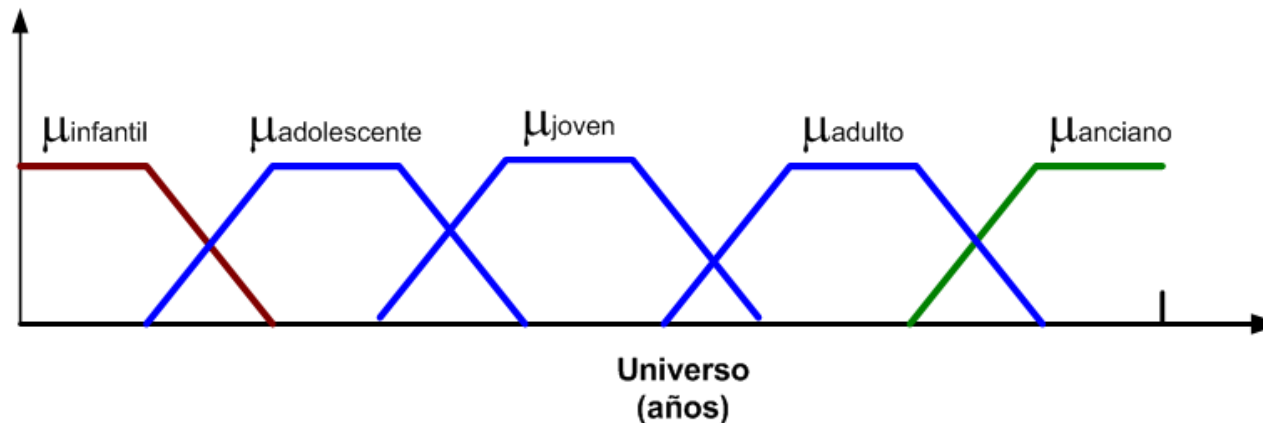
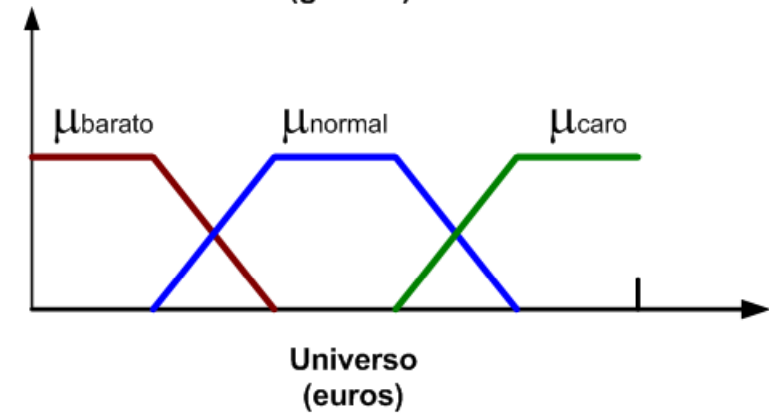
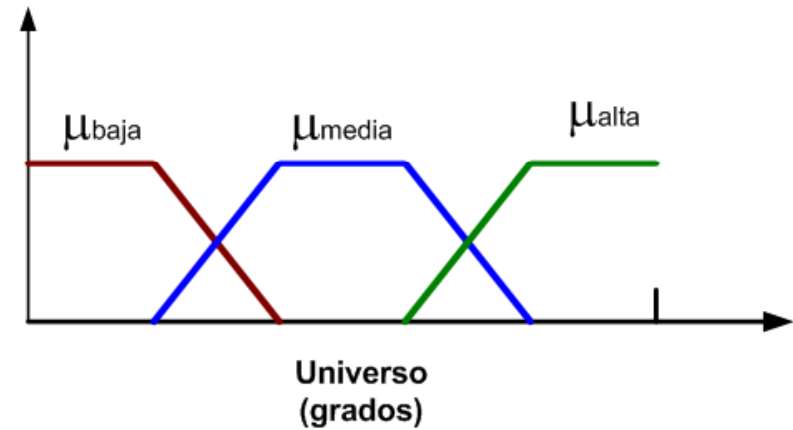
Estatura: {baja, media, alta}

Temperatura: {baja, media, alta}

Precio: {barato, normal, caro}

Edad: {infantil, adolescente, joven, adulto, anciano}

Etc.



Una **variable lingüística puede tomar diversos valores**, sin resultar contradictorio:

$$\text{Variable-Lingüística} = \text{Valor-difuso1}, \text{ Valor-difuso2}, \dots, \text{ Valor-difuso3}$$

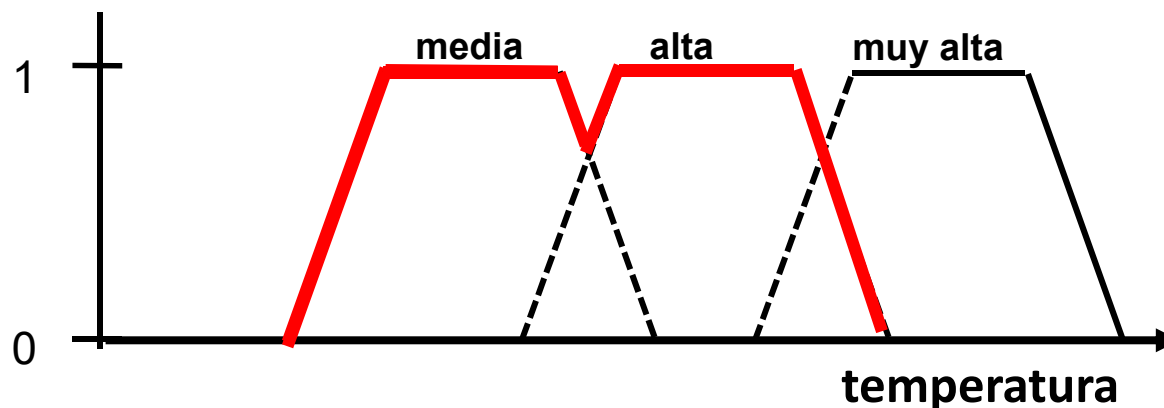
De esta manera, estamos asignando (OR) diversas distribuciones de posibilidad de valores en el dominio.

Ejemplo:

Como resultado de sucesivos procesos inferenciales, podemos obtener los siguientes valores sobre la variable lingüística temperatura:

$$\text{Temperatura} = \text{Alta}, \quad \text{Temperatura} = \text{Media} \Rightarrow \text{Temperatura} = \text{Alta} \cup \text{Temperatura} = \text{Media}$$

Estamos asignado distintas 'distribuciones de posibilidad' sobre el valor que puede tomar la temperatura:

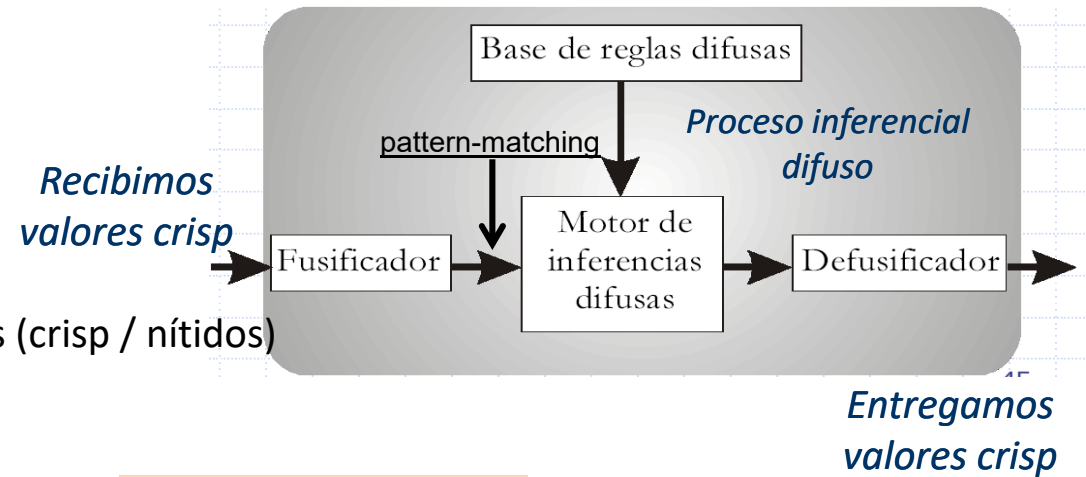


De acuerdo a esta distribución de posibilidades podremos obtener el valor de la temperatura (defusificación)

Inferencia Difusa

El proceso de Inferencia Difusa tiene 3 pasos:

1. Fusificación a valores difusos, y proceso de **pattern-matching**
2. Aplicación de Reglas Difusas
3. Defusificación o traducción a valores reales (crisp / nítidos)



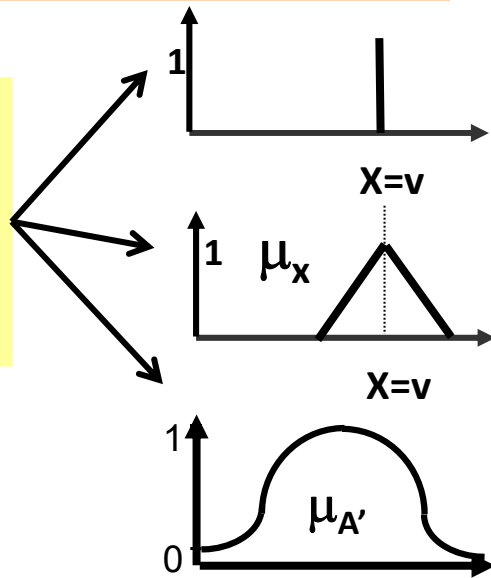
1) Fusificación / Pattern-Matching

Hecho

$x = \text{Valor}$

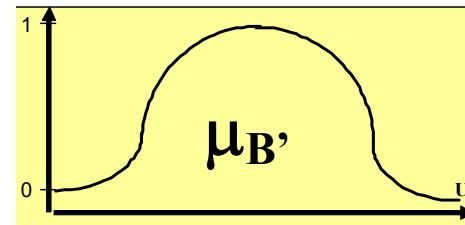
$x \in \{v_1 \dots v_n\}$

x es A' ($\mu_{A'}$)



2) Inferencia Difusa

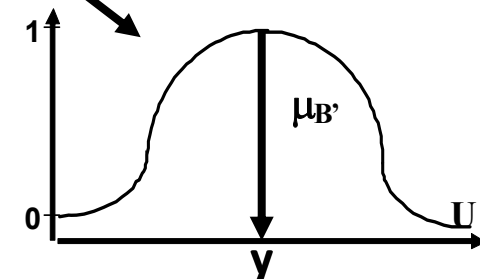
"si X es A_u entonces Y es B_v "



Modus Ponens difuso:
Matriz de Inferencia
Composicional

3) Defusificación

Medidas de Posibilidad
y Necesidad



Ejemplo:

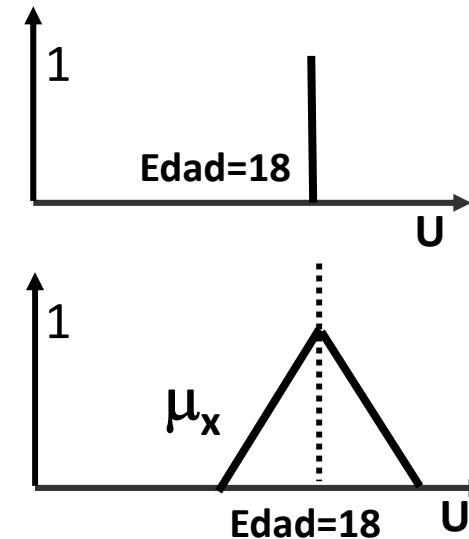
Agua:Fría \Rightarrow Apertura_Válvula:Poca

Si Agua:Tibia, ¿cuánto abrimos la válvula?

Fusificación de valores:

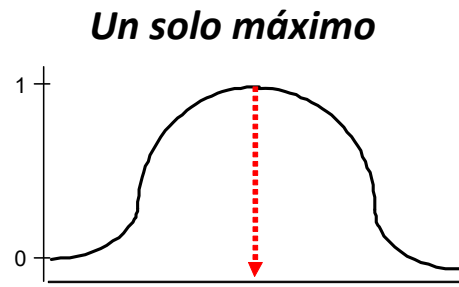
- a) **Fusificador tipo “singleton”:** Mapea un número real ($x=v$), o valor crisp, en conjunto difuso tipo “singleton”
- b) **Fusificador tipo “no singleton”:** Mapea un número real ($x=v$) en un conjunto difuso con una función de pertenencia (normalizada) definida. *En fuzzyclips (fuzzify edad 18 0.1)*

Con ello se puede realizar el pattern-matching de un valor crisp (fusificado) con conjuntos difusos.

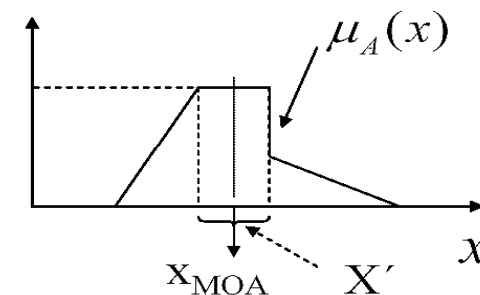


Defusificación:

- a) **Medio de los Máximos:**
(maximum-defuzzify ?var)

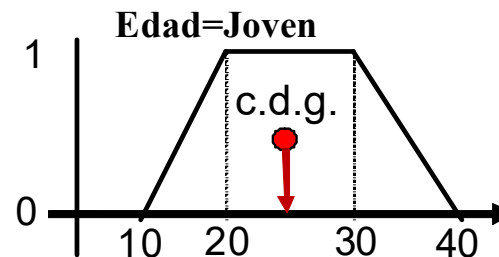


(Media de) Varios máximos



- b) **Método del Centro de Gravedad**
(moment-defuzzify ?var)

$$x_{COA} = \frac{\int x \mu_A(x) dx}{\int \mu_A(x) dx}$$



Inferencia Difusa: Modus Ponens Generalizado (MPG)

A, A' definido sobre U ; B, B' definido sobre V ,

$$X \text{ es } A' (\mu_{A'}) , \quad X \text{ es } A (\mu_A) \Rightarrow Y \text{ es } B (\mu_B)$$

$$Y \text{ es } B' (\mu_{B'})$$

Etapas:

- 1) Obtención de la **Matriz de Inferencia Composicional** (μ_R) de la regla difusa.

Se define sobre el producto cartesiano $U \times V$, representando la distribución condicional de:
"si x es A entonces Y es B ":

$$\mu_R: (U \times V) \rightarrow [0, 1] \quad \mu_R(u, v) = \min(\mu_A(u), \mu_B(v)), \quad \forall (u, v) \in U \times V$$

Toma valores en los universos U (premisa A) y V (conclusión B) y devuelve valores en $[0, 1]$

- 2) Obtención de la **función de pertenencia** $\mu_{B'}(v)$ de la conclusión, aplicando una *Regla Composicional de Inferencia* ($\mu_{B'}(v) = \mu_{A'}(u) \otimes \mu_R(u, v)$):

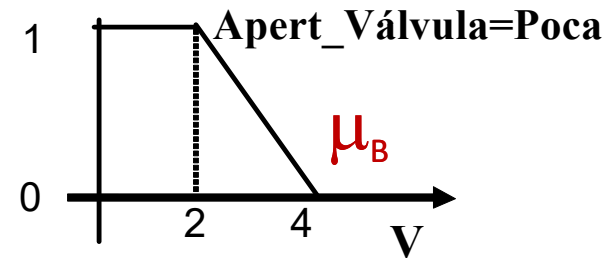
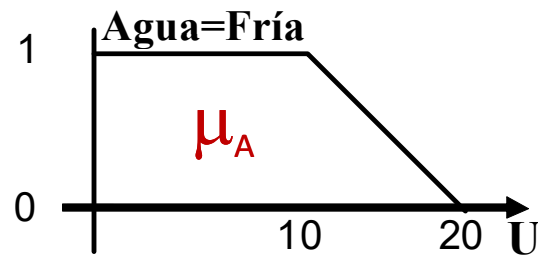
Existen varios métodos. Uno de los más utilizados es la **Regla de Inferencia max-min**:

$$\mu_{B'}(v) = \max(\min(\mu_{A'}(u), \mu_R(u, v)))$$

Toma valores en $[0, 1]$ y el universo U (dato A') y devuelve en universo V (Conclusión)

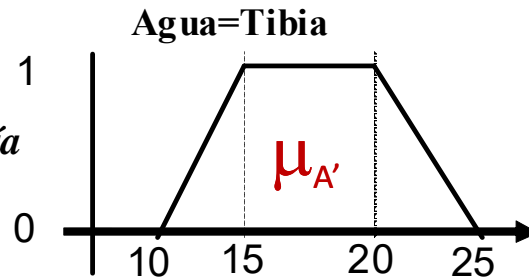
FuzzyClips: (*set-fuzzy-inference-type Max-min / Max-prod*)

Ejemplo-1: Si Agua:Fría \Rightarrow Apertura_Válvula: Poca



DATO

$\cong \text{Agua} = \text{Fría}$



$X \text{ es } A' (\mu_{A'}), \quad X \text{ es } A (\mu_A) \Rightarrow Y \text{ es } B (\mu_B)$

$Y \text{ es } B' (\mu_{B'})$

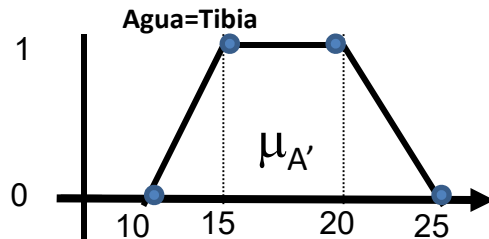
1) Obtención de $\mu_R(u,v)$

$$\mu_R(u, v) = \min(\mu_A(u), \mu_B(v)), \quad \forall (u,v) \in U \times V$$

Toma valores
en U (premisa A)
y V (conclusión B)
y devuelve valores
en $[0, 1]$

(v)	4	0	0	0	0	0
	3	0.5	0.5	0.5	0	0
	2	1	1	0.5	0	0
	1	1	1	0.5	0	0
	0	1	1	0.5	0	0
		0°	10°	15°	20°	25°
		(u)				

2) Obtención de $\mu_{B'}(v) = \max(\min(\mu_{A'}(u), \mu_R(u,v)))$



4	0	0	0	0	0
3	0.5	0.5	0.5	0	0
2	1	1	0.5	0	0
0	1	1	0.5	0	0
(v)	0º	10º	15º	20º	25º

$\mu_R(u,v)$

(u)

$$\mu_{B'}(0) = \max \{ \min(\mu_{A'}(0), \mu_R(0,0)), \min(\mu_{A'}(10), \mu_R(10,0)), \min(\mu_{A'}(15), \mu_R(15,0)), \min(\mu_{A'}(20), \mu_R(20,0)), \min(\mu_{A'}(25), \mu_R(25,0)) \} =$$

$$\max \{ \min(0,1), \min(0,1), \min(1,0.5), \min(1,0), \min(0,0) \} = 0,5$$

$$\mu_{B'}(2) = \max \{ \min(0,1), \min(0,1), \min(1,0.5), \min(1,0), \min(0,0) \} = 0,5$$

$$\mu_{B'}(3) = \max \{ \min(0,0.5), \min(0,0.5), \min(1,0.5), \min(1,0), \min(0,0) \} = 0,5$$

$$\mu_{B'}(4) = \max \{ \min(0,0), \min(0,0), \min(1,0), \min(1,0), \min(0,0) \} = 0$$

— Valores de agua=tibia

— Valores en matriz de inferencia

Aplicados sucesivamente para temperatura = 0, 10, 15, 20, 25

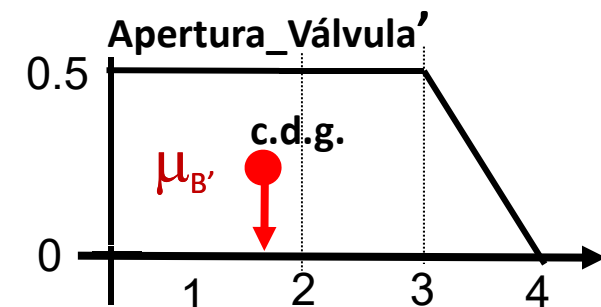
3) Defusificación:

Conclusión:

Si el **agua está tibia**, abre la válvula:

1.5 puntos (*Maximum-defuzzify*),

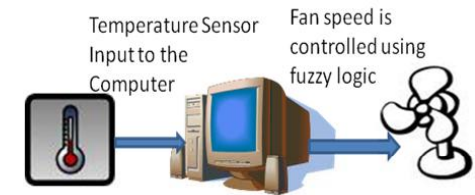
1.76 puntos (*Moment-defuzzify*).



Ejemplo-2: Control de un Termostato (*Control Difuso*)

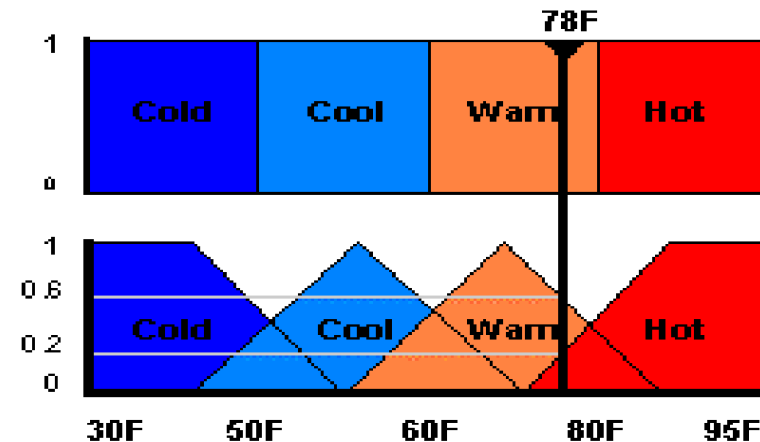
- Un controlador térmico tiene como entrada la **temperatura de la habitación** y como la salida el ajuste de la **velocidad del calentador**.
- Un termostato convencional trabaja como una llave de conexión **ON-OFF**: Si lo ponemos a **25°** el termostato se activa cuando la temperatura baja de **23°**; mientras que se apaga cuando alcanza **27°** (**histéresis**)

Como resultado, la temperatura de la habitación resulta **DEMASIADO CALIENTE** o **FRIA**.



Un **termostato difuso** trabaja ‘en tonalidades’, donde la temperatura es tratada como una serie de rangos que se superponen.

Por ejemplo: 26 °C (78 °F) es un 60% cálido y un 20% caliente.



El controlador difuso es programado con reglas **IF-THEN**, de forma que el ventilador varía gradualmente.

Como resultado, cuando la temperatura cambia, la velocidad del ventilador se ajusta gradualmente para mantener la temperatura al nivel deseado:

- | | |
|-----------------------------------|---------------------------------|
| R1. IF temperature IS cold | THEN fan_speed IS high |
| R2. IF temperature IS cool | THEN fan_speed IS medium |
| R3. IF temperature IS warm | THEN fan_speed IS low |
| R4. IF temperature IS hot | THEN fan_speed IS zero |

Fusificación.

El dato de temperatura de entrada (**78°F**) hace matching con las funciones de pertenencia:

$$\begin{aligned}\text{cold}(78^\circ\text{F}) &= 0 \\ \text{cool}(78^\circ\text{F}) &= 0 \\ \text{warm}(78^\circ\text{F}) &= 0.6 \\ \text{hot}(78^\circ\text{F}) &= 0.2\end{aligned}$$

Inferencia:

Las reglas son evaluadas.

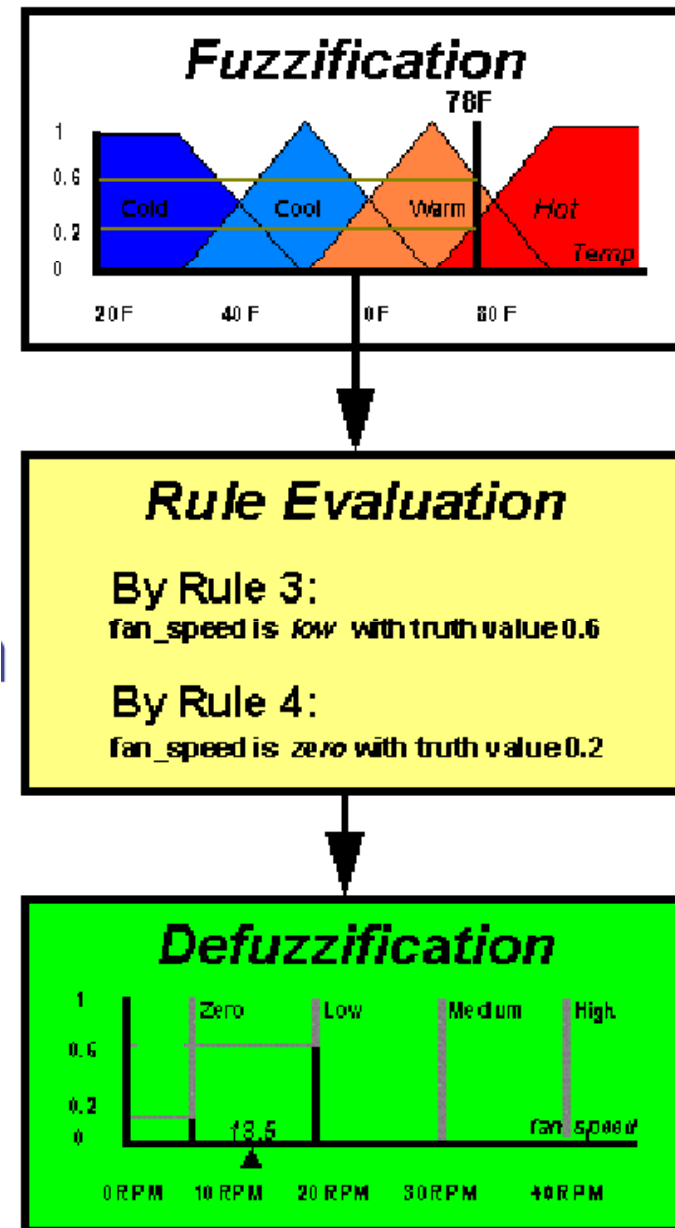
Para **78 °F** solo **2** reglas son activadas.

- **R3**, velocidad low = 60%.
- **R4**, velocidad zero = 20%.

*Asertamos
dos valores a
la variable
difusa!*

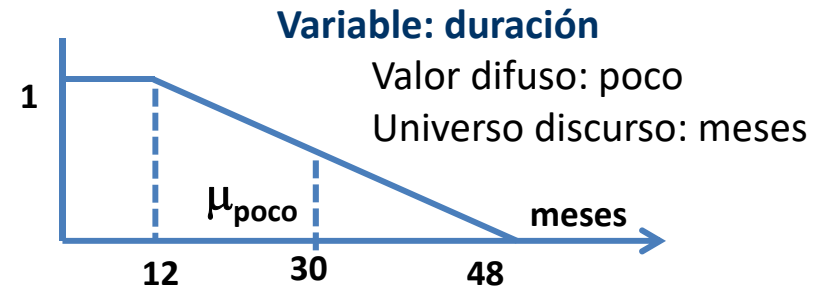
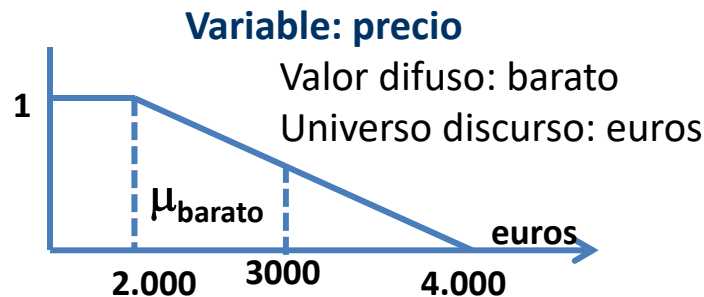
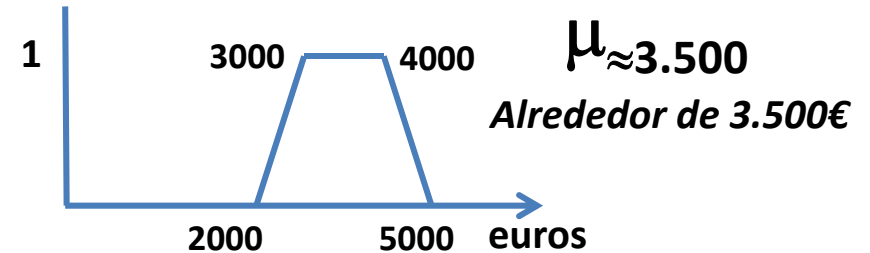
Defusificación:

La ponderación **60%** y el **20%** de los valores “low” y “zero”, respectivamente, son combinados y producen una salida numérica de **13.5 RPM** para la velocidad del calentador.



Ejemplo-3: Si el precio de un objeto es barato entonces su tiempo de duración es poco

Dato: El objeto cuesta alrededor de 3.500 euros,
¿Qué puedo decir sobre su duración?



1) Obtención de $\mu_R(u,v)$

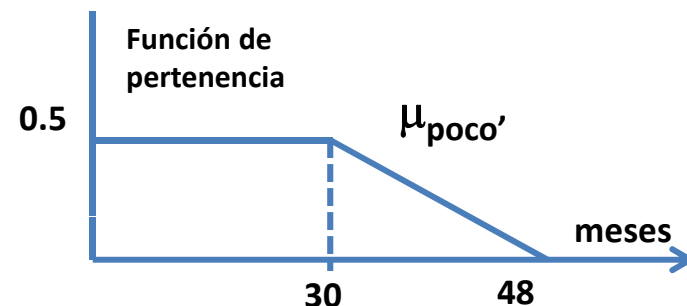
$$\mu_R(u, v) = \min(\mu_A(u), \mu_B(v)), \quad \forall (u,v) \in U \times V$$

48	0	0	0	0	0
30	0.5	0.5	0.5	0	0
12	1	1	0.5	0	0
0	1	1	0.5	0	0
(v)	0	2000	3000	4000	5000

Matriz de inferencia composicional

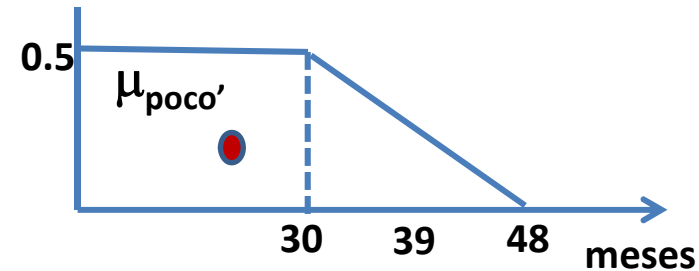
2) Obtención de $\mu_{poco'}(v)$

$$\mu_{B'}(v) = \max(\min(\mu_{A'}(u), \mu_R(u,v)))$$



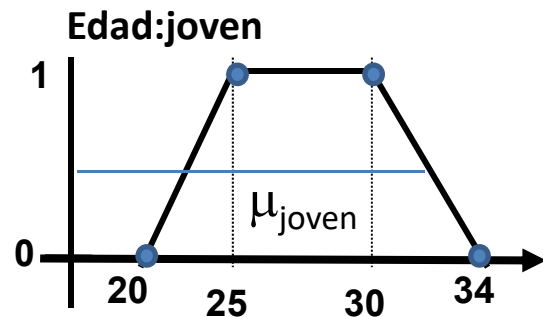
**La duración de un objeto que
cuesta alrededor de 3.500 euros es:**

Si defusificamos, la duración estimada serían:
15 meses (maximum), 18.66 (moment)



Una función de pertenencia μ_A induce una distribución de posibilidades:

- a) Posibilidad de que siendo μ_A sea $x=V$: $\mu_A(V)$
- b) Posibilidad de que siendo μ_A sea $x \in C$: $P(C, \mu_A) = \max [\mu_A(x)]_{\forall x \in C}$
- c) Necesidad de que siendo μ_A , sea $x \in C$: $N(C, \mu_A) = 1 - P(\neg C, \mu_A) = 1 - \max[\mu_A(x)]_{\forall x \notin C}$

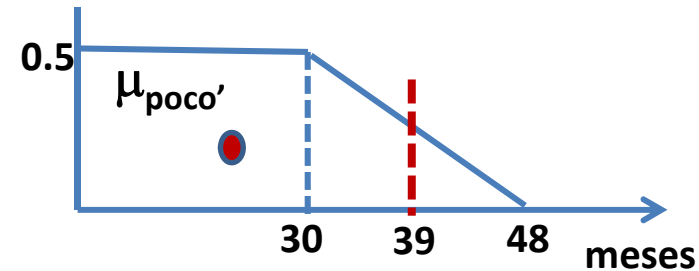


- a) Posibilidad de que siendo joven, tenga 25 años: $\mu_{joven}(25) = 1$
- b) Posibilidad de que siendo joven, tenga más de 32 años: $\max[\mu_{joven}]_{x>32} = 0.5$
- c) Necesidad de que siendo joven, tenga más de 32 años:

$$N(x>32, \mu_{joven}) = 1 - P(x \leq 32, \mu_{joven}) = 1 - \max[\mu_{joven}(x)]_{x \leq 32} = 1 - 1 = 0$$

**La duración de un objeto que
cuesta alrededor de 3.500 euros es:**

Si defusificamos, la duración estimada serían:
12 meses (maximum), 18.66 (moment)



Una función de pertenencia μ_A induce una distribución de posibilidades:

- a) Posibilidad de que siendo μ_A sea $x=V$: $\mu_A(V)$
- b) Posibilidad de que siendo μ_A sea $x \in C$: $P(C, \mu_A) = \max [\mu_A(x)]_{\forall x \in C}$
- c) Necesidad de que siendo μ_A , sea $x \in C$: $N(C, \mu_A) = 1 - P(\neg C, \mu_A) = 1 - \max[\mu_A(x)]_{\forall x \notin C}$

a) Posibilidad de que un objeto de 'alrededor de 3.500' euros dure '39 meses': $\mu_{poco'}(39) = 0.25$

b) Posibilidad de que un objeto de 'alrededor de 3.500 euros' dure 'más de 39 meses':

$$P(x > 39, \mu_{poco'}) = \max [\mu_{poco'}(x)]_{x > 39} = 0.25$$

c) Necesidad de que un objeto de 'alrededor de 3.500' euros dure 'más de 39 meses' :

$$N(x > 39, \mu_{poco'}) = 1 - P(x \leq 39, \mu_{poco'}) = 1 - \max[\mu_{poco'}(x)]_{x \leq 39} = 1 - 0.5 = 0.5$$

La regla A (precio=barato) \rightarrow B (duración=poco) solo vale para inferir la 'poca-duración' de un objeto, en la medida de 'lo barato' que es. Pero:

** Cuanto más vale, no se sabe que dure más. * Cuando menos vale, no se sabe que dure menos.*

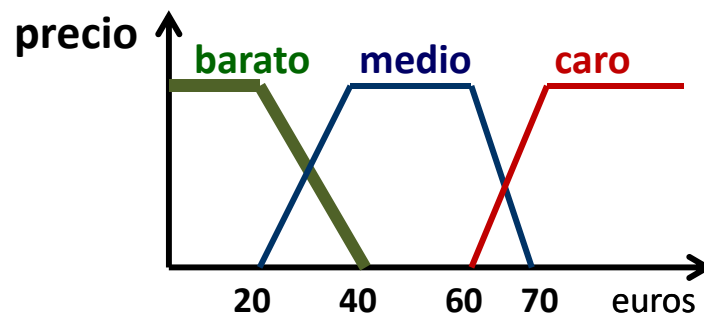
Simplemente, en la medida que sea 'de precio barato', inferiremos sobre su poca-duración.

Implementación en Fuzzy-Clips

Declaración de los conjuntos difusos:

```
(deftemplate precio ;Variable difusa
  0 100 euros ;Universo
  ( (barato (20 1) (40 0)) ;Valores difusos
    (medio (20 0) (40 1) (60 1) (70 0))
    (alrededor-treinta (25 0) (30 1) (35 0))
    (caro (60 0) (70 1))
    (veinte-cinco (25 0) (25 1) (25 0))
    (treinta (30 0) (30 1) (30 0))
    (treinta-cinco (35 0) (35 1) (35 0))))
```

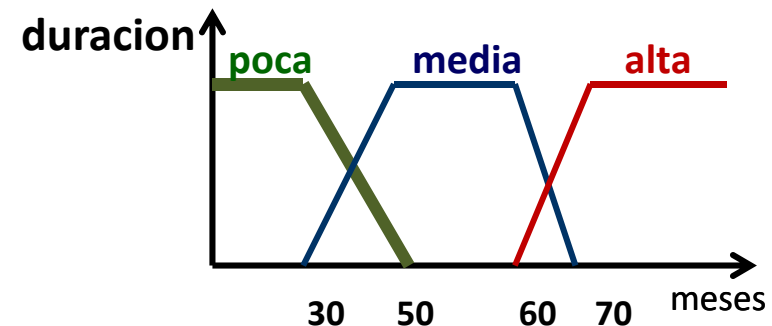
```
(deftemplate duracion ;Variable difusa
  0 100 meses ;Universo
  ( (poca (30 1) (50 0)) ;Valores difusos
    (media (30 0) (50 1) (60 1) (70 0))
    (alta (60 0) (70 1))))
```



Reglas:

```
(defrule estimacion
  (precio barato)
=>
  (assert (duracion poca)))
```

```
(defrule defuzzificar
  (declare (salience -1))
  (duracion ?dur)
=>
  (printout t "Se estima que va a durar "
    (maximum-defuzzify ?dur) " meses" crlf))
```



(assert (precio barato))

(run)

Se estima que va a durar 15 (20.4) meses

(assert (precio alrededor-treinta))

(run)

Se estima que va a durar 19 (22.1) meses

(assert (precio medio))

(run)

Se estima que va a durar 20 (22.6) meses

(assert (precio caro))

(run) *no matching*

(*) entre paréntesis, defusificando con moment-defuzzify

Nota: Si se aserta que el precio es 'barato' o 'medio' el sistema infiere la máxima información sobre 'duracion=poca':

(assert (precio barato))

(assert (precio caro)) >> (precio [barato OR caro])

(run)

Se estima que va a durar 15 (20.4) meses

Ya que tenemos una única regla con la premisa 'precio=barato', no pudiendo inferir sobre 'precio=caro'.

(assert (precio treinta))

(run)

Se estima que va a durar 20 (22.6) meses

(assert (precio treinta-cinco))

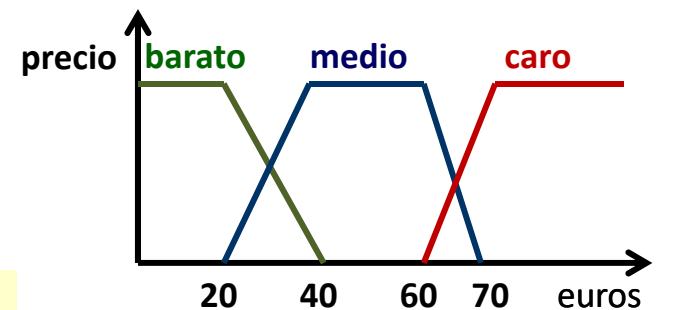
(run)

Se estima que va a durar 22.5 (23.8) meses

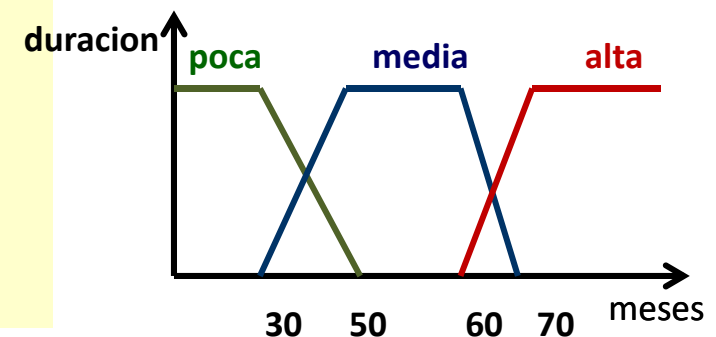
(assert (precio veinte-cinco))

(run)

Se estima que va a durar 17.5 (21.45) meses



(precio barato) => (duracion poca)



Ahora, supongamos las reglas:

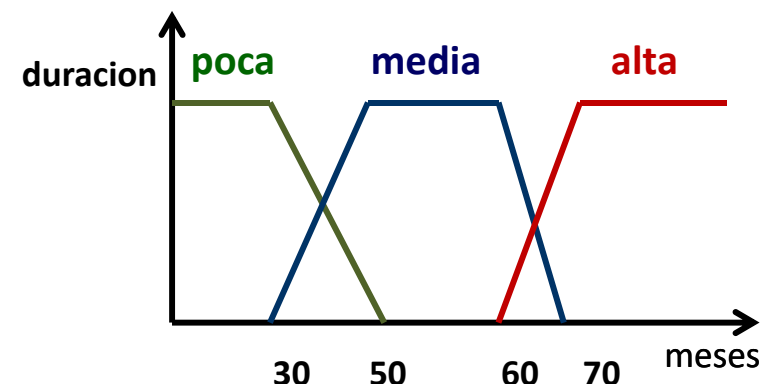
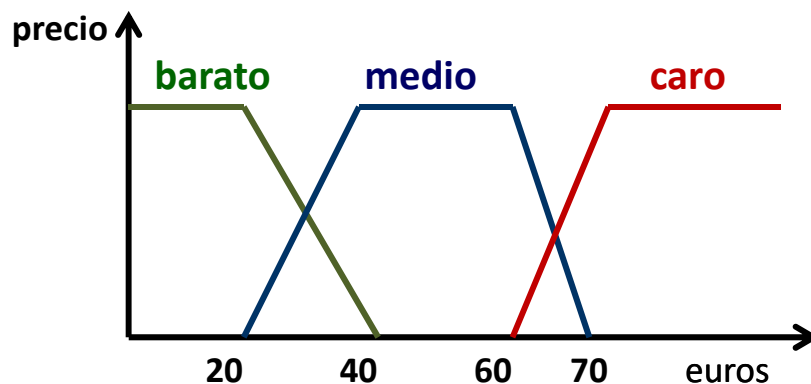
(defrule estimacion-barata
(declare (salience 100))
 (precio barato)
=> (assert (duracion poca)))

(defrule estimacion-medio
(declare (salience 100))
 (precio medio)
=> (assert (duracion media)))

(defrule estimacion-carro
(declare (salience 100))
 (precio caro)
=> (assert (duracion alta)))

PRECIO	DURACION (ahora)	DURACION (antes)
(assert (precio barato))	15	15
(assert (precio alrededor-treinta))	19	19
(assert (precio medio))	55	20
(assert (precio caro))	85	-
(assert (precio treinta))	32.5	20
(assert (precio treinta-cinco))	53.75	22.5
(assert (precio veinte-cinco))	35.63	17.5
(assert (precio barato)) (assert (precio medio))	35	15

Un dato puede hacer matching con más de una regla



En el razonamiento difuso, nueva información o nuevas inferencias se añaden (no reemplazan) a los valores previos de las variables difusas:

```
FuzzyCLIPS> (reset)
FuzzyCLIPS> (assert (precio alrededor-treinta))
<Fact-1>
FuzzyCLIPS> (assert (precio very caro))
<Fact-2>
FuzzyCLIPS> (assert (precio more-or-less barato))
<Fact-3>
FuzzyCLIPS> (assert (precio somewhat medio))
<Fact-4>
FuzzyCLIPS> (fuzzify precio 35 0.5)
<Fact-5>
```

```
Facts (MAIN)
f-0      (initial-fact) CF 1.00
f-5      (precio ???) CF 1.00
( (20.0 1.0) (22.0 0.9655) (24.0 0.9283) (26.0 0.8879) (28.0 0.8434)
  (29.00 0.8165) (30.0 1.0) (31.2 0.7595) (31.39 0.7541) (32.0 0.7746)
  (34.0 0.8367) (34.93 0.8636) (35.0 1.0) (35.07 0.8675) (36.0 0.8944)
  (38.0 0.9487) (40.0 1.0) (60.0 1.0) (61.0 0.9487) )
```

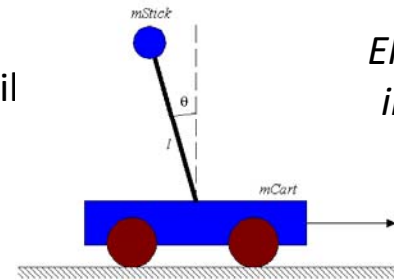
```
FuzzyCLIPS> (run)
Se estima que va a durar 51.666666666666666 meses
```

*Defusificar al final del
razonamiento*

```
Facts (MAIN)
f-0      (initial-fact) CF 1.00
f-5      (precio ???) CF 1.00
f-9      (duracion [ [ [ poca ] OR [ media ] ] OR [ alta ] ] OR [ poca ] ) CF 1.00
( (30.0 1.0) (40.0 0.5) (50.0 1.0) (60.0 1.0) (65.0 0.5) (32.0 0.7746)
  (70.0 1.0) ) (34.93 0.8636) (35.0 1.0) (35.07 0.8675) (36.0 0.8944)
  (38.0 0.9487) (40.0 1.0) (60.0 1.0) (61.0 0.9487) )
```

Ejemplo-4: Péndulo Invertido

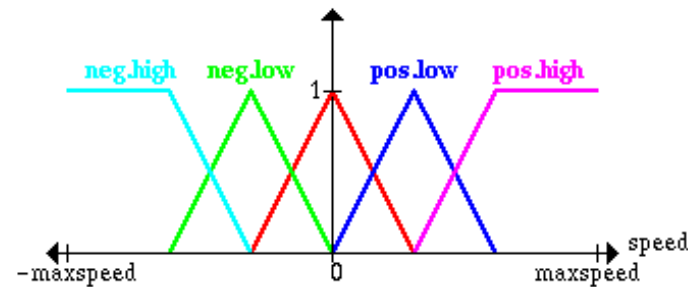
El problema es balancear un poste en una plataforma móvil que puede moverse en solamente dos direcciones, a la izquierda o a la derecha.



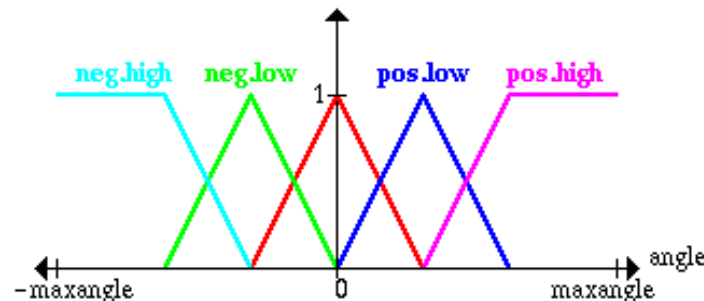
El control de un péndulo invertido fue una de las primeras aplicaciones relevantes de la inferencia difusa.

Valores Difusos:

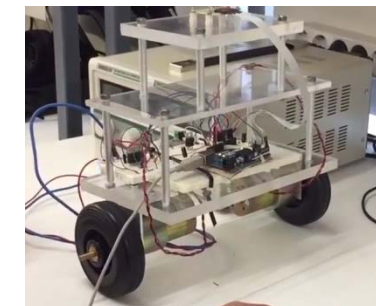
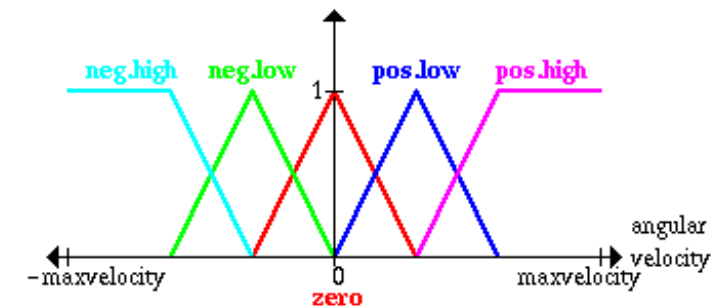
Velocidad de la plataforma (speed):



Angulo del poste:

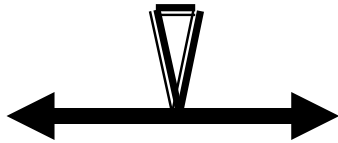


Velocidad angular del poste:



Self balancing robot with fuzzy logic

Reglas:



Ángulo / Velocidad	NH	NL	Z	PL	PH
NH	NH	NH	NH	NH	PH
NL	NH	NL	NL	Z	PH
Z	NH	NL	Z	PL	PH
PL	NH	Z	PL	PL	PH
PH	NH	NL	PH	PH	PH

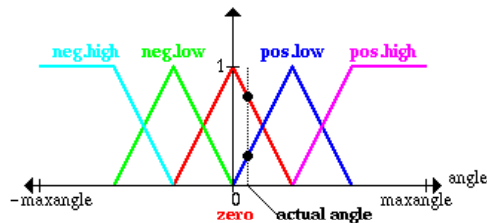
NH Negative High
NL Negative Low
PL Positive Low
HL Positive High

Ejemplos:

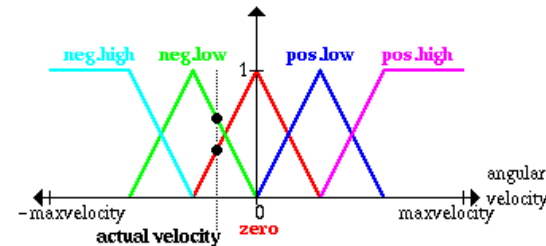
R1: If angle is zero **and** angular-velocity is zero **then** speed shall be zero

R2: If angle is zero **and** angular-velocity is pos-low **then** speed shall be pos-low

Consideremos un caso concreto



El ángulo es muy poco positivo: $A=Z$, $A=PL$



La velocidad es muy poco negativa: $V=Z$, $V=PL$

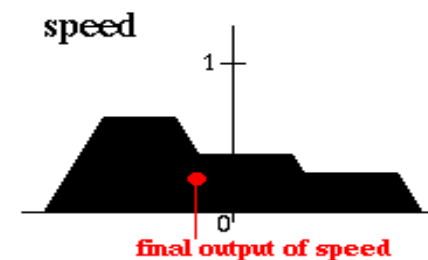
VIDEO

VIDEO

VIDEO

Podemos aplicar varias reglas, y combinando las conclusiones, obtenemos la velocidad de la plataforma:

$Angle=Z, Velocity=Z \rightarrow Speed=Z$,
 $Angle=Z, Velocity=PL \rightarrow Speed=PL$,
 $Angle=PL, Velocity=Z \rightarrow Speed=PL$,
 $Angle=PL, Velocity=PL \rightarrow Speed=PL$



Características de la Inferencia Difusa

- **La pertenencia de un dato a más de un conjunto difuso (valor de variable difusa) es posible: no es contradictorio como en lógica clásica, monovaluada. Nuevos hechos difusos se acumulan a los conocidos.**
- **Una inferencia difusa no proporciona nunca conclusiones más precisas que B.**
 - Dado un hecho (X es A') y una regla difusa (*Si X es A , entonces $Y=B$*), podemos deducir información acerca de Y .
 - Si A' es ligeramente diferente de A , entonces obtendremos X es B' , siendo B' ligeramente diferente a B .
 - Nunca podremos obtener algo más preciso de X es B' .
- **Es útil aún cuando A' difiere ligeramente de A (contrariamente al Modus Ponens Clásico).**

El resultado de la inferencia se ve afectado por un nivel de indeterminación. Si A' es sensiblemente distinto de A , el MPG puede no ser suficiente para concluir un B' que nos aporte información fiable (se busca B).
- **A veces es necesaria mayor información sobre la relación causal entre X e Y alrededor de los conceptos A , B .**

Por ejemplo, a partir de la regla *"si el tomate es rojo entonces está maduro"* y del hecho *"el tomate es muy rojo"*,

el MPG no permite concluir *"el tomate está muy maduro"*,
a menos que se explicita que el grado de madurez aumenta con la intensidad del color rojo.

Aplicaciones Razonamiento Difuso:

Un sistema informático es difuso cuando su funcionamiento se basa, al menos en parte, en la lógica difusa.

- Estabilizadores imágenes en cámaras fotográficas, acondicionamiento aire, control temperatura, etc.
- Control de horno de cemento (Dinamarca).
- Control de horno de fundición (NKK Fukoyama)
- Acondicionador de aire doméstico (Mitsubishi)
- Lavadora (Viessmann, Fagor).
- Cámara autofocus (Canon)
- Fotocopiadora (Sanyo).
- Control de spam (Mozilla).
- Expert for *stock exchange activities* (Yamaichi, Hitachi)
- Cruise-control for *automobiles* (Nissan, Subaru)
- Back light control for *camcorders* (Sanyo)
- Recognition of handwritten symbols with *pocket computers* (Sony)
- Reconocedor de habla (NTT Japón)
- Sistema experto en medicina (Univ. De California)
- Control de robot autónomo (SRF, USA).

Existe cierta discusión sobre si el razonamiento bayesiano extendido es capaz de captar y razonar con información imprecisa

Entornos y Aplicaciones

- **FuzzyClips (desarrollado por National Research Council, Canadá)**
- European Centre for Soft Computing (<http://www.softcomputing.es/>)
- Repository for Fuzzy Logic (<http://www.fuzzytech.com/>)
- Fuzzy Logic Toolbox (<https://es.mathworks.com/products/fuzzy-logic.html>)

Conclusiones.

- No siempre se dispone de conocimiento totalmente preciso:
 - La información (hechos) de partida es imprecisa: hace mucho frío, es bastante bajo, tarda aproximadamente 10 minutos, está poco maduro, etc.
 - Las reglas tampoco son totalmente precisas (además de probablemente inciertas)
- Hay que trabajar con lógica difusa (tipos, variables y reglas difusas) e inferir nueva información sobre ella. Esto implica los siguientes pasos:
 - Fusificación: mapping a dominio difuso
 - Inferencia difusa: aplicación de reglas y razonamiento
 - Defusificación: mapping a dominio real para su aplicación
- La lógica difusa ayuda a tomar decisiones de forma similar a como lo hacen los humanos, donde no todo es conocido al 100%
 - Aplicación en multitud de sistemas que implican algún tipo de sistema de control: sensores, lavadoras, cámaras digitales, robots de limpieza, etc.

Ejercicio. Control termostato difuso

Implementar el controlador difuso de un ventilador, tal que controle la velocidad de un ventilador a partir de la temperatura y humedad de la habitación.

La tabla para determinar la **velocidad** en función de la temperatura y la humedad es:

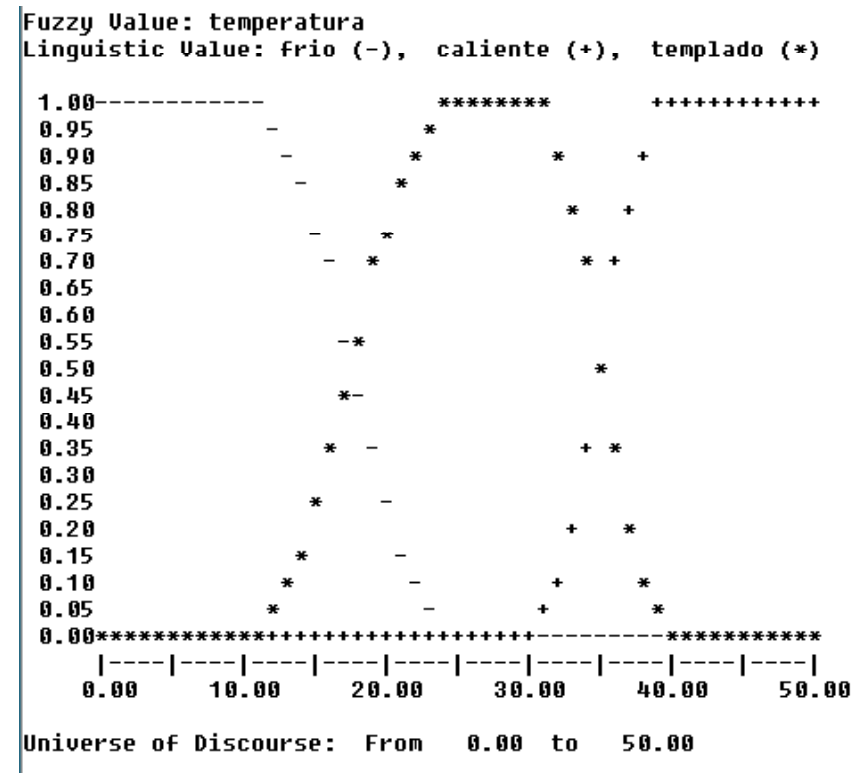


Temperatura/Humedad	seco	húmedo	mojado
frío	medio	alto	alto
templado	bajo	medio	alto
caliente	medio	alto	alto

- Definir adecuadamente las variables difusas: **temperatura, humedad, y velocidad**.
- Mediante un razonamiento difuso, obtener valores de velocidad para diferentes escenarios en la habitación.
- Finalmente, defisificar el valor de la velocidad para obtener el valor críps correspondiente.

Temperatura/Humedad	seco	húmedo	mojado
frío	medio	alto	alto
templado	bajo	medio	alto
caliente	medio	alto	alto

Variables Difusas: temperatura, humedad, y velocidad

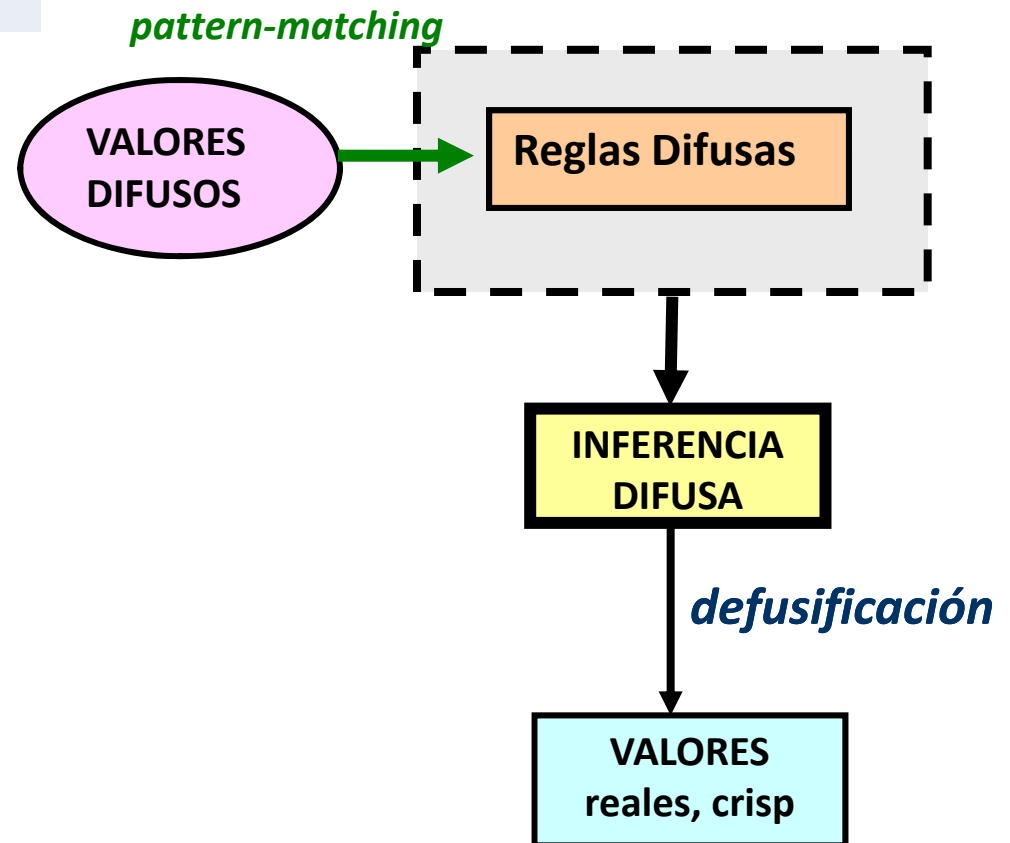


Temperatura/Humedad	seco	húmedo	mojado
frío	medio	alto	alto
templado	bajo	medio	alto
caliente	medio	alto	alto

REGLAS

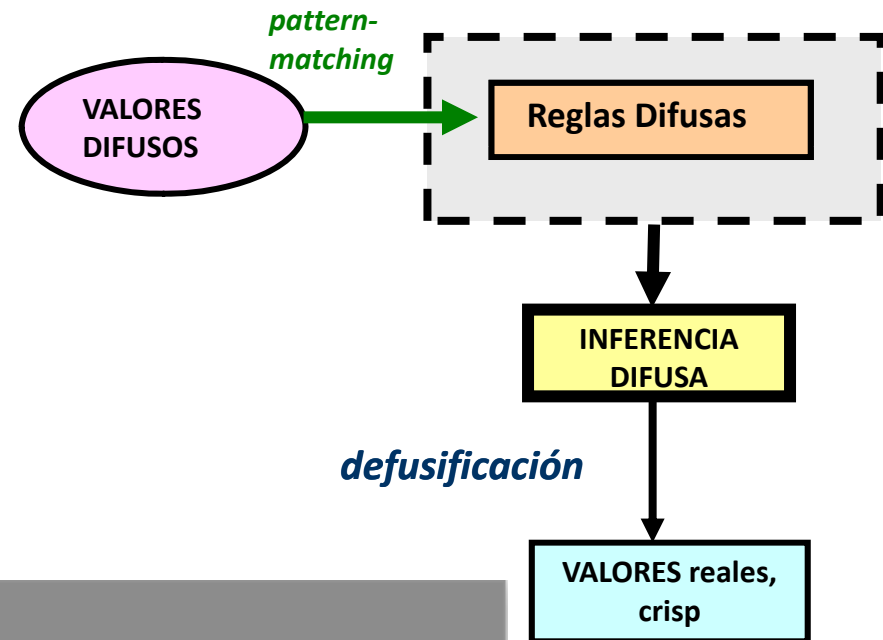
Temperatura/Humedad	seco	húmedo	mojado
frío	medio	alto	alto
templado	bajo	medio	alto
caliente	medio	alto	alto

DEFUSIFICACIÓN *(al final)*



En Resumen

Temperatura/Humedad	seco	húmedo	mojado
frío	medio	alto	alto
templado	bajo	medio	alto
caliente	medio	alto	alto



EJECUCIÓN

- (assert (temperatura frio))
- (assert (humedad mojado))

```

Facts (MAIN)
0      (initial-fact) CF 1.00
1      (temperatura frio) CF 1.00
2      (humedad mojado) CF 1.00 .75 0.875) (15.63 0.7188) (17.5 0.5)
( (70.0 0.0) (73.75 0.03125) (77.5 0.125) (81.25 0.2813) (85.0 0.5)
  (88.75 0.7188) (92.5 0.875) (96.25 0.9688) (100.0 1.0) )

```

- (run)

```

Facts (MAIN)
0      (initial-fact) CF 1.00
1      (temperatura frio) CF 1.00
2      (humedad mojado) CF 1.00 .75 0.875) (15.63 0.7188) (17.5 0.5)
6      (velocidad ???) CF 1.00 77.5 0.125) (81.25 0.2813) (85.0 0.5)
7      (crisp velocidad-moment-rpm 58.7421875) CF 1.00 (15.0 0.5)
8      (crisp velocidad-maximum-rpm 100.0) CF 1.00 9688) (100.0 1.0) )

```


Variables Difusas: temperatura, humedad, y velocidad

REGLAS

```
(deftemplate temperatura ;Variable difusa
  0 50 grados ;Universo
  ((frio (z 10 25))
   (caliente (s 30 40))
   (templado not [ frio or caliente ])))
```

```
(deftemplate humedad ;Variable difusa
  0 100 porcentaje ;Universo
  ((seco (z 0 30))
   (mojado (s 70 100))
   (humedo not [ seco or mojado ])))
```

```
(deftemplate velocidad ;Variable difusa
  0 100 rpm ;Universo
  ((bajo (z 0 30))
   (alto (s 70 100))
   (medio not [ bajo or alto ])))
```

```
(defrule temp-frio_hum-seco
  (temperatura frio)
  (humedad seco)
  =>
  (assert (velocidad medio)))
```

```
(defrule temp-templado_hum-seco
  (temperatura templado)
  (humedad seco)
  =>
  (assert (velocidad bajo)))
```

```
(defrule temp-templado_hum-humedo
  (temperatura templado)
  (humedad humedo)
  =>
  (assert (velocidad medio)))
```

```
(defrule temp-caliente_hum-seco
  (temperatura caliente)
  (humedad seco)
  =>
  (assert (velocidad medio)))
```

```
(defrule temp-frio_hum-humedo
  (temperatura frio)
  (humedad humedo)
  =>
  (assert (velocidad alto)))
```

Etc.....

DEFUSIFICACIÓN

```
(defrule defuzzificar-moment
  (declare (salience -100)) ;MINIMO PESO
  (velocidad ?val)
  => (assert (crisp velocidad-moment-rpm (moment-defuzzify ?val))))
```

```
(defrule defuzzificar-maximum
  (declare (salience -100)) ;MINIMO PESO
  (velocidad ?val)
  => (assert (crisp velocidad-maximum-rpm (maximum-defuzzify ?val))))
```