

## Práctica Control difuso con Qtfuzzylite

[Qtfuzzylite](#) es actualmente la herramienta de implementación de sistemas de control difuso que permite mayor funcionalidad. Su uso tiene ventajas e inconvenientes, que explicamos a continuación.

En cuanto a las ventajas, si comparamos Qtfuzzylite con la Fuzzy Toolbox que ofrece Matlab, vemos que mientras QtFuzzylite permite el uso de cualquier función lineal a trozos como función de pertenencia, la Fuzzy Toolbox de Matlab solo permite implementar funciones triangulares y trapezoidales. Además, Qt Fuzzylite, a partir de la versión 6.0, permite también entradas difusas y encadenamiento de reglas (no permitido por ningún otro software).

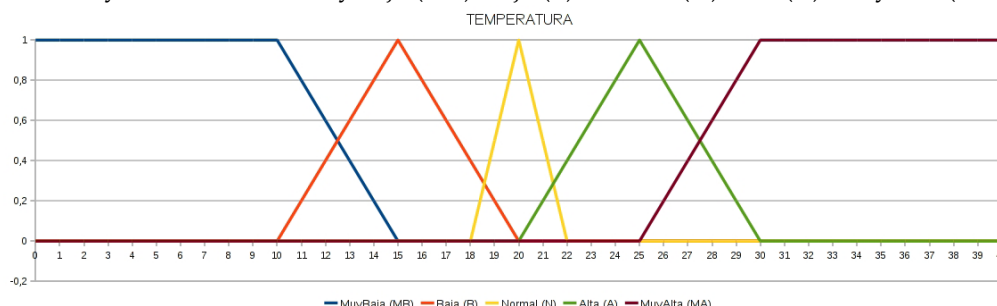
Respecto a los inconvenientes, QTFuzzylite es de pago. Sin embargo, gracias a [nuestra contribución en el desarrollo de la versión actual](#), el autor nos ha cedido para uso académico de los estudiantes de la asignatura la versión 4.0, que puedes descargar e instalar desde el Campus Virtual. Esta versión presenta algunos pequeños errores de redondeo, por lo que siempre habrá que verificar que los resultados que ofrece son correctos. A pesar de dichos errores, la herramienta puede ser de gran ayuda a la hora de resolver los problemas propuestos en la asignatura, pues permite visualizar los valores de los  $z_i$  (con los que activar las reglas) y las funciones de pertenencia de las sucesivas inferencias.

### TUTORIAL

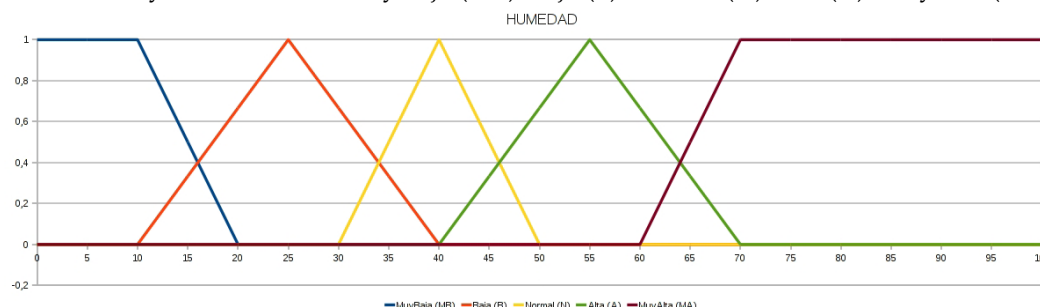
Para este tutorial utilizaremos el siguiente ejemplo:

*Control difuso de una calefacción*<sup>1</sup>. Se pretende controlar la calefacción de un invernadero, a partir de los parámetros de temperatura y humedad del mismo [1], [2]. Se cuenta con un sensor de temperatura con un rango de funcionamiento desde 0 a 40 grados centígrados, y una precisión de centésimas, y con un sensor de humedad con un rango de funcionamiento entre 0% y 100% de humedad relativa, con una precisión de centésimas. La caldera se puede controlar mediante incrementos/decrementos de temperatura, desde -15 a +15 grados centígrados. Se define para ello dos *variables de entrada o estado* (Temperatura y Humedad) y una *variable de salida* (Variación de temperatura):

- Temperatura, cuyos valores son: muy baja (MB), baja (B), normal (N), alta (A), muy alta (MA).



- Humedad, cuyos valores son: muy baja (MB), baja (B), normal (N), alta (A), muy alta (MA).



<sup>1</sup> Ejemplo obtenido de <https://ccia.esei.uvigo.es/docencia/IA/1213/transparencias/ejemplo-control-difuso.pdf>, posiblemente adaptado de Becker, M. et al: Fuzzy control for temperature and humidity in refrigeration systems. Proceedings of IEEE International Conference on Control and Applications, 1994.

- Variación de temperatura, cuyos valores son: bajada grande (BG), bajada normal (BN), bajada pequeña (BP), mantener (M), subida pequeña (SP), subida normal (SN), subida grande (SG):



Vamos a introducir este ejemplo en Qtfuzzylite.

### Fase 1. Introducción del problema

Para ello comenzamos dando un nombre al sistema de control (en este caso lo hemos llamado Control de Temperatura). Podemos ahora añadir una nueva variable de entrada o salida. Las variables de entrada corresponden a las que figuran en los antecedentes de las reglas, y debes definirlas en la casilla destinada a los Inputs (pulsando en el botón con la cruz). De igual modo las variables de salida corresponden a las que figuran en los consecuentes de las reglas, y debes definirlas en la casilla destinada a los Outputs. **Ten cuidado y define cada tipo de variable en la casilla adecuada, ya que si no lo haces así no podrás corregirlo y deberás introducir la variable de nuevo.** Al pulsar el botón de la cruz, nos sale el interfaz para añadir variables. Añadimos la variable Temperatura en Inputs.

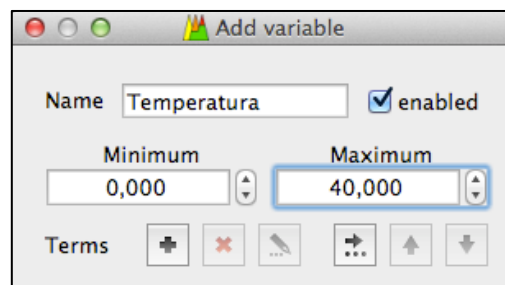
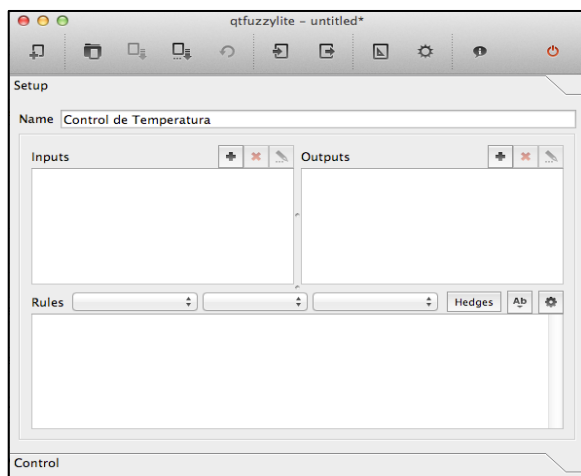


Figura 1. Nombre del sistema e introducción de la variable Temperatura

Definimos el universo de discurso de la variable mediante la introducción de los valores mínimo y máximo que puede tomar, y pasamos a añadir sus valores lingüísticos, pulsando en la cruz que aparece junto a la palabra Terms.

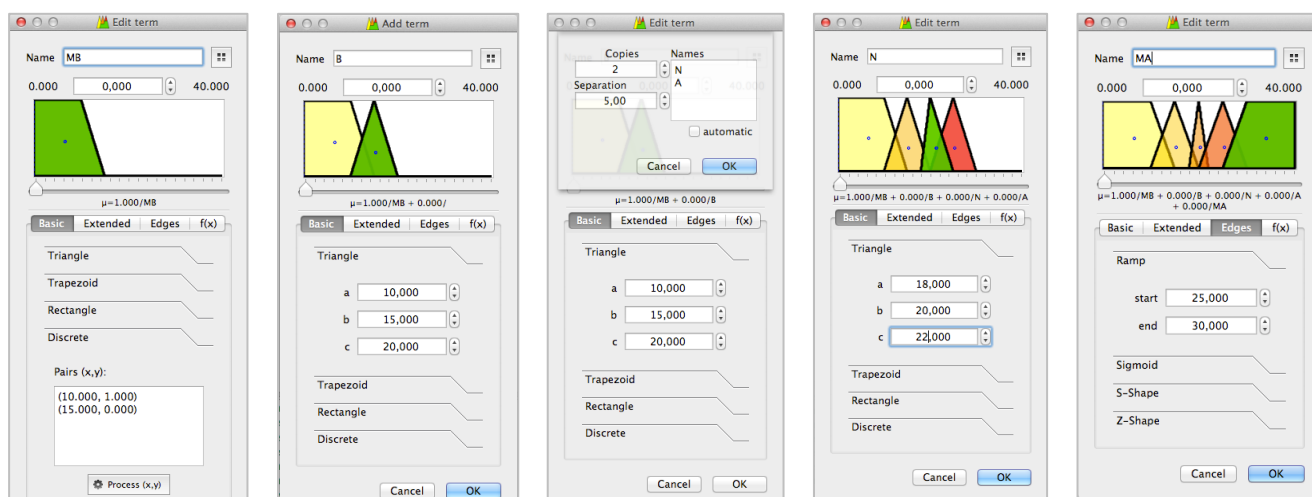


Figura 2. Definición de los valores lingüísticos de la variable Temperatura

En cada pestaña nos salen las opciones más comunes para definir los valores lingüísticos. De igual modo se introducen las variables Humedad (en Inputs) y Variación de temperatura (en Outputs):

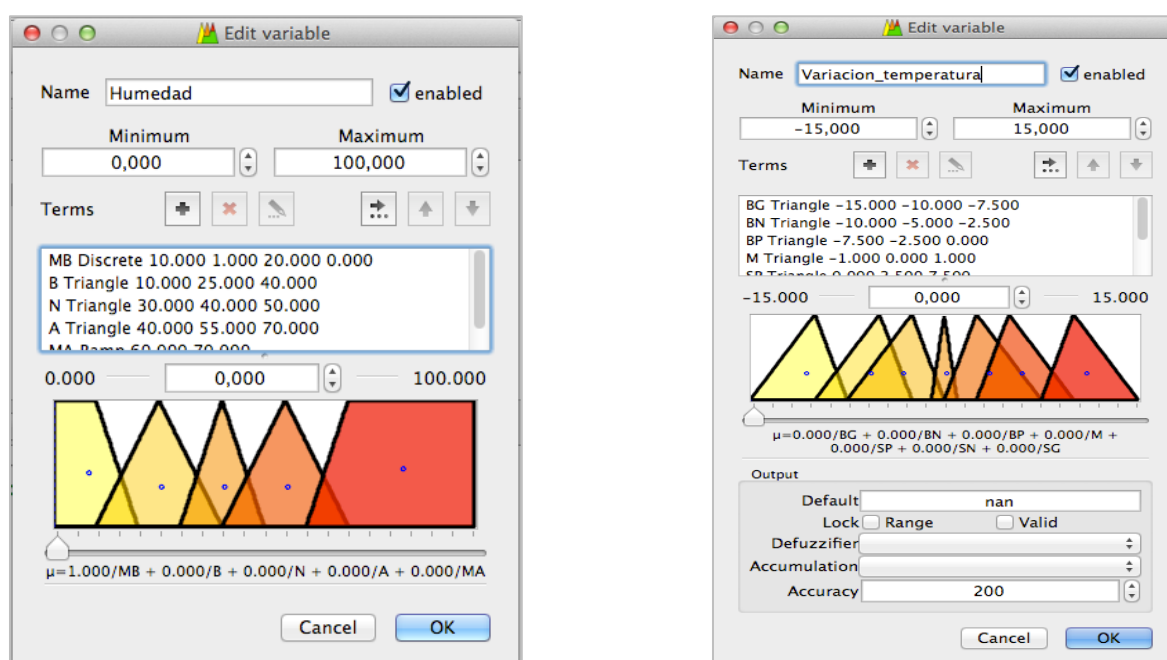


Figura 3. Definimos las variables Humedad y Control de Temperatura

Para introducir ahora las reglas, podemos pulsar el botón y se generan todas las combinaciones posibles de antecedentes (con AND) ya preparadas para editarlas. Supongamos que tenemos la siguiente *matriz de asociación* para la variable de control Variación de temperatura.

	Humedad				
Temperatura	MB	B	N	A	MA
MB	SN	SN	SG	SG	SG
B	M	M	SP	SP	SN
N	M	M	M	M	BP
A	M	M	BP	BP	BN
MA	BP	BN	BN	BG	BG

Editamos las reglas (la figura no las muestra todas). Para construir las reglas podemos escribir las “palabras” que podemos ver al pulsar el botón **Hedges** (etiquetas o modificadores). Si en las reglas escribimos alguno(s) de ello(s), se irán marcando automáticamente en la lista los que utilizemos.

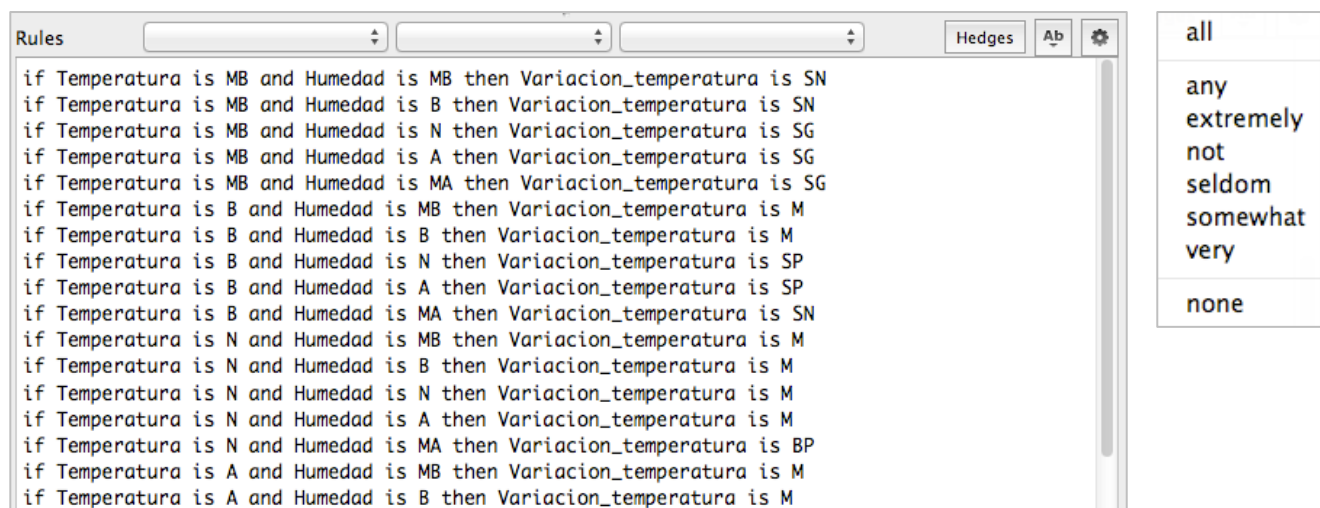



Figura 4. Edición de reglas y etiquetas disponibles en “Hedges”

Una vez definido el conjunto de reglas, mediante el botón  las procesamos. Las reglas correctas se colorean de verde y las incorrectas de rojo (por ejemplo, en este caso el valor SH no está definido para la variable Variación de temperatura).

```
if Temperatura is MB and Humedad is MB then Variacion_temperatura is SH
#[variable error] term <SH> not found in variable <Variacion_temperatura> {/src/variable/Variab
if Temperatura is MB and Humedad is B then Variacion_temperatura is SN
if Temperatura is MB and Humedad is N then Variacion_temperatura is SG
if Temperatura is MB and Humedad is A then Variacion_temperatura is SG
```

Figura 5. Aviso de error

Corregimos los errores y volvemos a procesar las reglas, hasta que estén correctas todas:

```
#-----
# Total rules: 25. Good Rules: 25. Bad Rules: 0.
# Rules successfully processed at 9:13.12 am (12/05/14)
# You may proceed to control the engine
```

Control

Figura 6. Reglas correctas y permiso para acceder a la parte de control (en azul)

Ya podemos acceder a la parte de control, pulsando en la pestaña correspondiente (justo abajo del mensaje en azul).

## Fase 2. Control difuso

Para poder ahora realizar razonamientos difusos, debemos antes indicar a la herramienta qué funciones debe utilizar como t-norma (para el AND), t-conorma (para el OR), método de activación (cualquier t-norma), decodificador (defuzzifier), y acumulación (cualquier t-conorma). La t-norma, la t-conorma y el método de activación se eligen en los tres primeros desplegables que aparecen junto a “Rules” (en el orden indicado). Poniéndonos encima de cada desplegable, sale el nombre de la función a introducir.

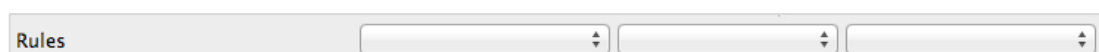


Figura 7. Introducción de t-norma, t-conorma y método de activación

Para la t-norma y el método de activación disponemos de las siguientes funciones:

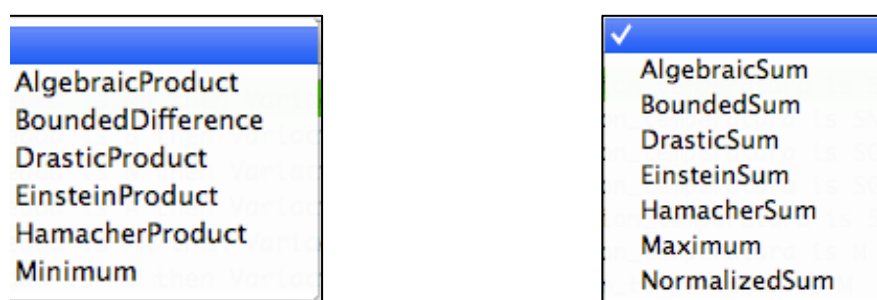


Figura 8. Métodos disponibles: para t-normas y activación; para t-conormas y acumulación

El método de acumulación y el de decodificación se eligen para cada variable de salida. Si no lo hicimos al definir la variable, lo podemos hacer ahora, editando de nuevo la variable en la pestaña Setup:

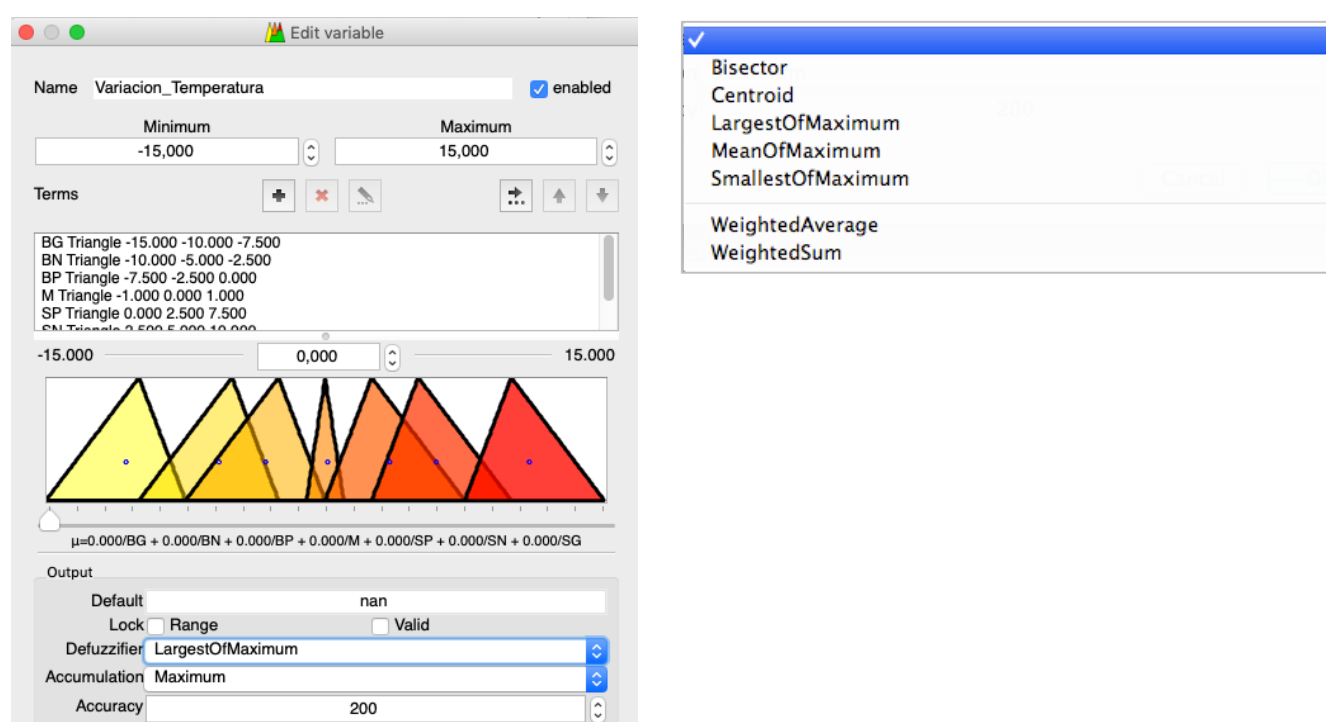


Figura 9. Definición de métodos de acumulación/decodificación, y opciones para decodificar

Vemos que en esta ventana se nos ofrecen otras opciones:

- Default: Es un valor que usará el sistema cuando no se activa ninguna regla o bien si la variable de salida está deshabilitada. Puede tener valores -inf, inf, nan (not a number).
- Lock range: asegura que la variable de salida siempre retornará valores entre los límites que esté definida
- Lock valid: devolverá el último valor válido obtenido para la variable de salida (mientras no se active ninguna regla, la salida no cambiará).

Una vez elegidas las cinco funciones, podremos razonar introduciendo valores nítidos para las variables de entrada en las casillas correspondientes o utilizando los deslizadores de cada variable. Supongamos que para nuestro ejemplo elegimos como t-norma la del mínimo (tendríamos entonces que elegir la t-conorma del máximo, para que se satisfagan las leyes de deMorgan) y método de activación, Mamdani (mínimo); como método de acumulación, el máximo y como decodificador, el mayor de los máximos.

A medida que deslicemos los valores de entrada podremos ver con qué conjuntos difusos hay coincidencia, y en qué grado (en la expresión " $\mu=...$ " que aparece bajo cada variable) y también como evoluciona el conjunto difuso de salida y su valor nítido (según el método de decodificación elegido).

Supongamos un valor nítido para Temperatura de 12 grados (pertenece al conjunto difuso MB con grado 0.6, y al B con grado 0.4) y un valor nítido para Humedad de 18 (pertenece a los conjuntos difusos MB y B, en grados 0.2 y 0.533, respectivamente). Por tanto, serán de aplicación las reglas 01, 02, 06 y 07). Tras aplicar las reglas se obtienen los conjuntos difusos marcados en verde, que tras el proceso de acumulación y decodificación con las funciones elegidas nos da un valor nítido de 7.275.

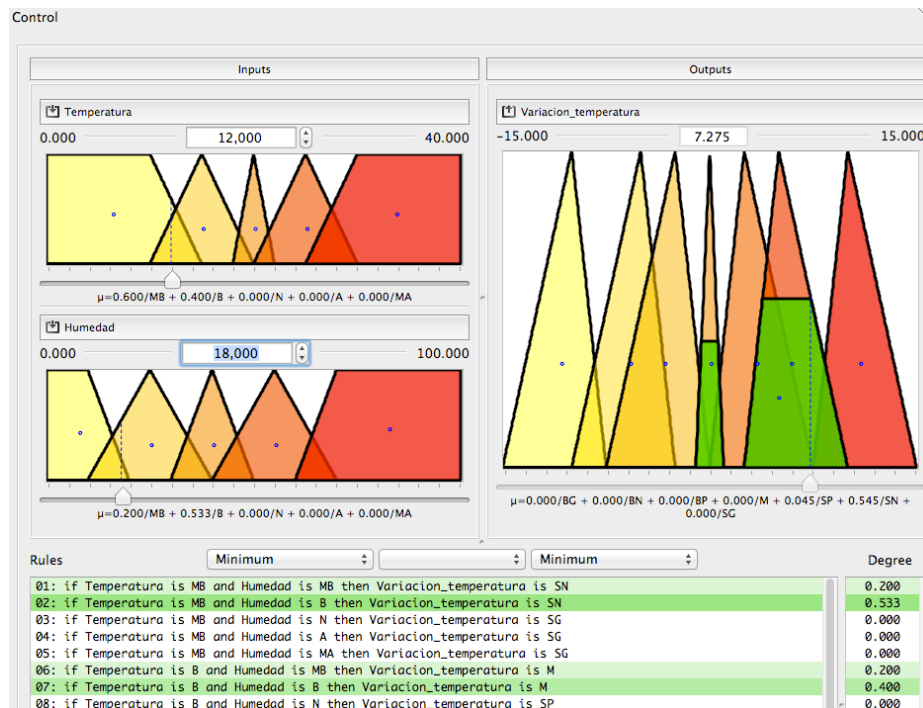


Figura 10. Razonamiento en qtfuzzylite

Además de las opciones tradicionales de guardar un problema o cargar uno existente, qtfuzzylite permite además opciones para importar o exportar el sistema de control difuso completo en varios formatos. Por ejemplo, si exportamos a C++:

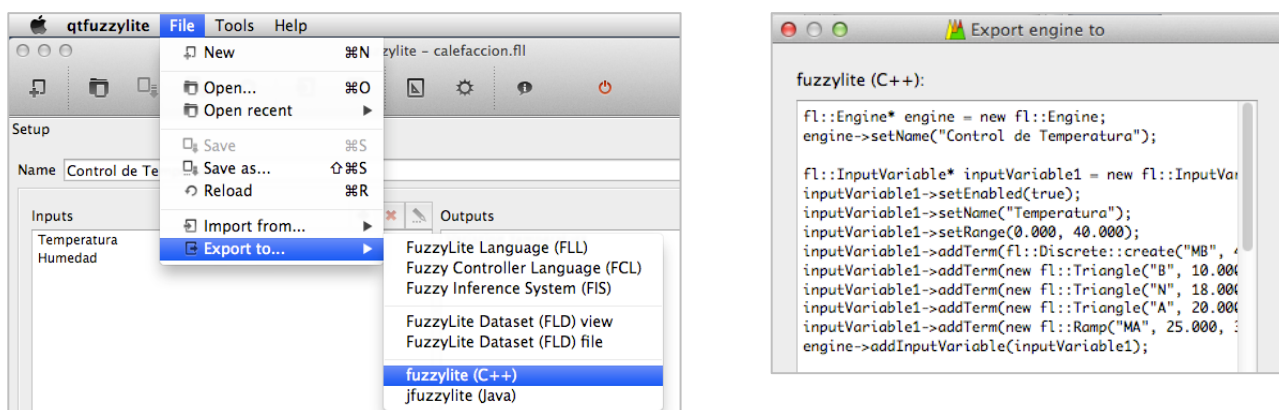


Figura 11. Guardar un problema, abrir, importar, ... (exportando a C++)

Contesta ahora a la **pregunta 1 (parte Tarea y entrega)**



## Encadenamiento de reglas con Qtfuzzylite<sup>2</sup>

La mayoría de herramientas de control difuso no permiten el encadenamiento de reglas, ni la entrada de conjuntos difusos en lugar de valores nítidos. Por ejemplo, ni Qtfuzzylite 4.0 ni la Fuzzy Toolbox de Matlab permiten ninguna de las dos cosas. A partir de la versión 6.0, Qtfuzzylite permite ambas.

Para poder apoyarnos en el software a la hora de resolver problemas de encadenamiento, utilizaremos un método que permite “simular” el funcionamiento de un sistema de control difuso con encadenamiento. Dicho método fue propuesto por Juan Rada, desarrollador de Qtfuzzylite, antes de implementar esta funcionalidad en versiones posteriores del software. Llamaremos al método “encadenamiento simulado”.

Para ilustrar el método, supongamos por ejemplo un problema en que una variable llamada “Presión” es a su vez salida de un conjunto de reglas y entrada de otro conjunto (véase la pregunta 3, en la parte “Tarea y entrega” de este enunciado). El método consiste en duplicar la variable Presión para poderla utilizar tanto de variable de entrada como de variable de salida. Para ello, definimos la variable “Presión” como variable de salida (de la primera cadena de reglas), y nuevamente como variable de entrada (de la segunda cadena de reglas), llamándola por ejemplo “Salida-Presión”. De este modo, podemos realizar la primera cadena de inferencias, y una vez obtenida la solución, trasladar el valor obtenido para la variable “Presión” y utilizarlo como valor de entrada de la variable “Salida-Presión” para poder realizar el siguiente conjunto de inferencias.

Es importante ser conscientes de que el método de encadenamiento simulado no suele producir el mismo resultado que en un sistema con encadenamiento real. El motivo es que, en un sistema real, se realiza la primera cadena de inferencias, obteniendo como resultado un conjunto difuso de salida A que se utiliza como la entrada de la segunda cadena de inferencias. Sin embargo, en el encadenamiento simulado, dicho conjunto A se decodifica con alguna técnica (centroide, media de máximos, etc.) para obtener un valor a, y es este valor a el que se utiliza como entrada de la segunda cadena de inferencias. La pérdida de información que se produce al decodificar el conjunto A y utilizar el valor a, hace que los resultados no tengan por qué coincidir. Por tanto, la salida del software debe tomarse como un apoyo visual a la correcta ejecución del problema.

### TAREA Y ENTREGA

**Tarea:** Contesta a las preguntas que figuran a continuación

**Entrega:** Documento pdf con la solución (capturas de pantalla y textos descriptivos)

**Pregunta 1.** En el problema del tutorial, considera ahora las siguientes funciones: para la unión, la suma drástica; para la intersección, el producto drástico; como método de activación, la función max-prod (en Qtfuzzylite se llama AlgebraicProduct) y como método de decodificación, el centroide. Se pide:

- Captura la última pantalla, y compara con los resultados con los obtenidos anteriormente.
- Repite este ejercicio considerando las siguientes entradas: Temperatura 12 y Humedad 8; Temperatura 12 y Humedad 70.
- Observa y comenta cómo cambia la forma del conjunto de salida en todas las opciones realizadas (tutorial, apartado a y apartado b). Explica en cada caso cual sería en tu opinión el mejor método para obtener un valor nítido a partir de cada conjunto de salida (*defuzzificar*).

Para los siguientes problemas, impleméntalos en Qtfuzzylite. Describe los pasos de la inferencia, incluyendo capturas de pantalla. Si la aplicación da errores respecto a la solución de clase, coméntalos.

**Pregunta 2.** La abuela María prepara sus deliciosas galletas caseras de forma artesanal desde hace más de 40 años. El toque secreto de la receta consiste en hornearlas cuidadosamente de acuerdo con las siguientes reglas:

- Si las galletas están un poco crudas, entonces la temperatura del horno debe ser alta.
- Si las galletas están medio hechas, entonces la temperatura del horno debe ser media.
- Si las galletas están doraditas, entonces la temperatura del horno debe ser baja.

Tras diversas entrevistas con la abuela se han podido establecer los siguientes conjuntos difusos sobre un índice cromático especial (0 = galleta cruda; 10 = galleta chamuscada) y la temperatura del horno:

<sup>2</sup> (utilizarás esta parte del tutorial para poder resolver los problemas 2 y 3).

Índice cromático de las galletas:

*un poco crudas:* (1/4, 0.5/6, 0/7)  
*medio hechas:* (0/3, 1/5, 1/6, 0/8)  
*doraditas:* (0/5, 1/7)

Temperatura del horno (°C):

*baja:* (0/150, 1/160, 1/180, 0/190)  
*media:* (0/170, 1/190, 1/210, 0/230)  
*alta:* (0/210, 1/220, 1/240, 0/250)

Supóngase que se interpretan las reglas anteriores como implicaciones de Mamdani. Suponiendo que en cierto momento el índice cromático de las galletas es 6, ¿cuál será el valor de temperatura aplicado al horno si se utiliza la técnica del primer valor máximo para obtener valores nítidos?

**Pregunta 3** (encadenamiento de reglas). Considérese un sistema con las siguientes reglas, interpretadas como implicaciones de Mamdani:

- R1. Si la temperatura es *alta* entonces la presión es *elevada*.
- R2. Si la temperatura es *baja* entonces la presión es *baja*.
- R3. Si la presión es *baja* entonces la entrada de combustible debe ser *grande*
- R4. Si la presión es *elevada* entonces la entrada de combustible debe ser *pequeña*

Con los siguientes conjuntos difusos:

Temperatura(°C):

*baja* = (0/0 0.2/30 0.8/40 1/50 0.7/60 0.2/70 0/80)  
*alta* = (0/50 0.3/60 0.8/70 1/80 1/90 0.5/100 0/110)

Presión(bar):

*baja* = (0/0 0.4/200 0.8/400 1/600 1/800 0.8/1000 0.4/1200 0/1400)  
*Elevada* = (0/1000 0.2/1200 0.4/1400 0.8/1600 1/1800 1/1900 0.5/2000 0/2200)

Entrada combustible(litros/hora):

*pequeña* = (0/0 0.6/1 1/2 1/3 0.4/4 0/5)  
*grande* = (0/4 0.5/5 1/6 0.5/7 0/8)

Si la temperatura actual es 60°C, determina el valor para la entrada de combustible empleando la técnica del primer valor máximo para transformar valores difusos en nítidos

**Pregunta 4** (encadenamiento de reglas y antecedentes compuestos). Agapito lleva 8 años encargado del control de la turbina GG-35 y es ya una autoridad en su manejo. Agapito ha reconocido que para controlar la turbina solamente se fija en el ruido que produce y en un sensor de temperatura:

- R1. Si el nivel de ruido es normal y la temperatura es alta, entonces establece una velocidad suave.
- R2. Si el nivel de ruido es normal y la temperatura no es alta, entonces establece una velocidad moderada.
- R3. Si el nivel de ruido es bajo, entonces establece una velocidad alta.
- R4. Si la velocidad es suave, la fuerza de frenado debe ser normal.
- R5. Si la velocidad es moderada, la fuerza de frenado debe ser alta.
- R6. Si la velocidad es alta, la fuerza de frenado debe ser alta.

Tras diversas entrevistas con Agapito se han elaborado los siguientes conjuntos difusos para los valores del nivel de ruido, temperatura, velocidad y fuerza de frenado:

Nivel de ruido (escala 0 a 12)

*bajo* (0/1, 1/3, 1/5, 0/7)  
*normal* (0/5, 1/7, 1/9, 0/11)

Temperatura (escala 20°-100° C):

*alta* (0/40, 1/60, 0/80)

Velocidad (escala 0 a 100 rpm)

*suave* (0/10, 1/30, 0/50)  
*moderada* (0/30, 1/50, 0/70)  
*alta* (0/40, 0.5/50, 0.5/60, 1/70, 0.5/80, 0.5/90, 0/100)

Fuerza de frenado (escala 0 a 5)

*normal* (0/1, 1/3, 0/5)  
*alta* (0/3, 1/4)

Suponiendo que en cierto momento el nivel de ruido es 5,5 y la temperatura de 50° C, calcula el valor de la fuerza de frenado si se utiliza la técnica de la media de los valores máximos para obtener valores nítidos.