**Taller 1 (punto 5)**

**Presentado por:**

* Camila Lozano Jiménez - código 20201020161.
* Juan Felipe Rodríguez Galindo - código 20181020158.

**Enunciado:**

Empleando los comandos de MATLAB proponer un sistema de lógica difusa (Mamdani) basado en reglas para la toma de decisiones. Se pueden considerar como referencia:

• La propina en un restaurante.

• Operación de una lavadora.

• El control de un horno para el pan.

**Configuraciones:**

• Conjuntos (funciones de pertenencia) en el antecedente: 3, 5 y 7.

• Conjuntos (funciones de pertenencia) en el consecuente: 3, 5 y 7.

**Requerimientos de diseño:**

• Propuesta original.

• Emplear los comandos de MATLAB (no la interfaz gráfica)

**Desarrollo**

1. **Tomar valores propuestos para el ejercicio:** En este caso, como los valores asignados para el grupo son C y B, se toma 7 y 5 como cantidades de antecedentes y consecuentes.
2. Para definir, dentro de los posibles estados, los de la retina que determinan la presencia de la retinopatía diabética tenemos las siguientes convenciones que serán nuestras variables de entrada:
   1. Espesor macular
   2. Quistes intrarretinianos
   3. Elipsoides
   4. Desorganización de las capas de la retina

Y la variable de salida, nivel de la enfermedad, estará determinada por las siguientes condiciones:

* 1. Temprana
  2. Avanzada
  3. Severa
  4. Atrófica

1. Se realiza el código de Matlab para determinar la presencia de la retinopatía diabética:

% Ejercicio de taller con base en el proyecto de detección de retinopatía

% dibética

% implemmentado en comandos

% Punto 5 Corte 1

close all

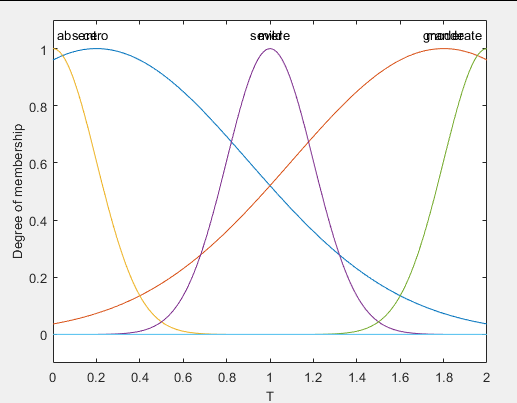
clear all

warning('off')

% Sistema

a = newfis('Retinopatía dibética')

Graficando se puede observar:



Luego se realiza el código en Matlab para cada una de las entradas:

% Variable de entrada : Thickness

a=addvar(a,'input','T',[0 2]); % Rango de espesor

% Funciones de pertenencia

a=addmf(a,'input',1,'cero','gaussmf',[0.7 0.2]);

a=addmf(a,'input',1,'grande','gaussmf',[0.7 1.8]);

plotmf(a,'input',1)

% Variable de entrada : Cyst

a=addvar(a,'input','Q',[0 3]); % Quistes metarretineanos

% Funciones de pertenencia

a=addmf(a,'input',1,'absent','gaussmf',[0.2 0]);

a=addmf(a,'input',1,'mild','gaussmf',[0.2 1]);

a=addmf(a,'input',1,'moderate','gaussmf',[0.2 2]);

a=addmf(a,'input',1,'severe','gaussmf',[0.2 3]);

plotmf(a,'input',1)

% Variable de entrada : EZ

a=addvar(a,'input','EZ',[0 2]); % Elipsoides

% Funciones de pertenencia

a=addmf(a,'input',1,'intact','gaussmf',[0.2 0]);

a=addmf(a,'input',1,'disrupted','gaussmf',[0.2 1]);

a=addmf(a,'input',1,'absent','gaussmf',[0.2 2]);

plotmf(a,'input',1)

% Variable de entrada : DRIL

a=addvar(a,'input','DRIL',[0 2]); % Desaorganización de las capas de la retina

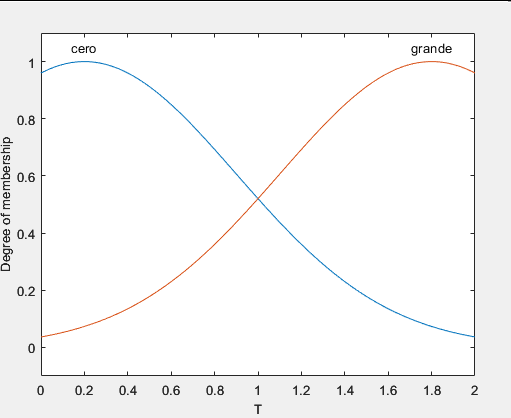
% Funciones de pertenencia

a=addmf(a,'input',1,'whitout','gaussmf',[0.7 0.2]);

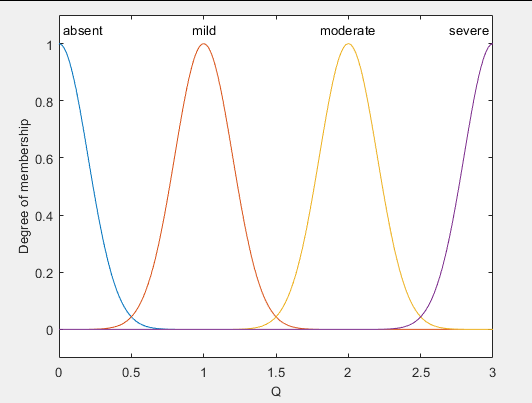
a=addmf(a,'input',1,'with','gaussmf',[0.7 1.8]);

plotmf(a,'input',1)

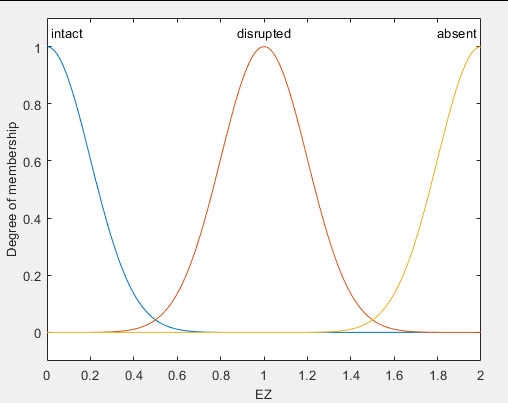
Graficando se puede observar:



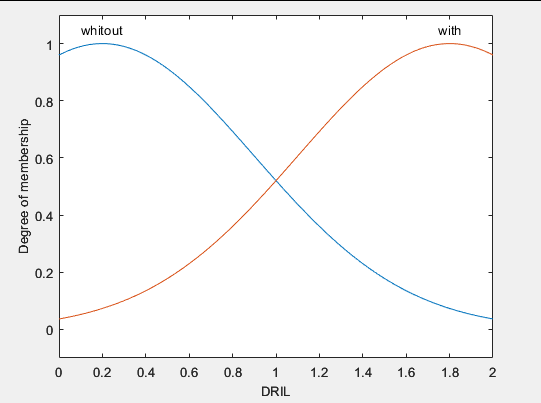
Thickness



Quist



EZ



DRIL

Finalmente, se realiza el nivel de retinopatía diabética

% Variable de salida : Retinopatía diabética

a=addvar(a,'input','nivel de retinopatia',[0 4]); % Quistes metarretineanos

% Funciones de pertenencia

a=addmf(a,'input',1,'absent','trimf',[0 0.5 1]);

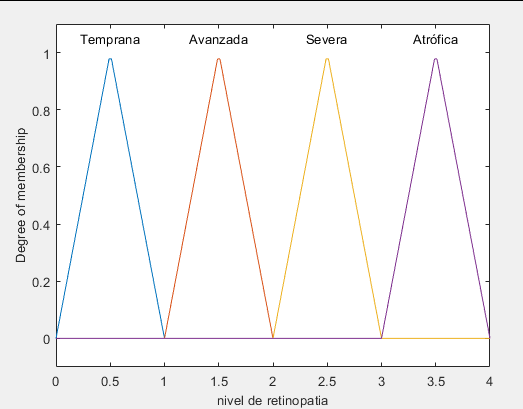
a=addmf(a,'input',1,'mild','trimf',[1 1.5 2]);

a=addmf(a,'input',1,'moderate','trimf',[2 2.5 3]);

a=addmf(a,'input',1,'severe','trimf',[3 3.5 4]);

plotmf(a,'input',1)

y se grafica de la siguiente manera:



Nivel de retinopatía.

Posteriormente, se definen las reglas de inferencia de la siguiente manera:

* Columna 1: Temprana
* Columna 2: Avanzada
* Columna 3: Severa
* Columna 4: Profunda

Estas reglas de inferencia se basaron en [An optical coherence tomography-based grading of diabetic maculopathy proposed by an international expert panel: The European School for Advanced Studies in Ophthalmology classification](https://pubmed.ncbi.nlm.nih.gov/31718271/), TABLA 2

Se definen las siguientes reglas por código:

ruleList = [

1 1 1 1 1 1 1 % Temprana

1 2 1 1 2 1 1 % Avanzada

2 2 1 2 3 1 1 % Severa

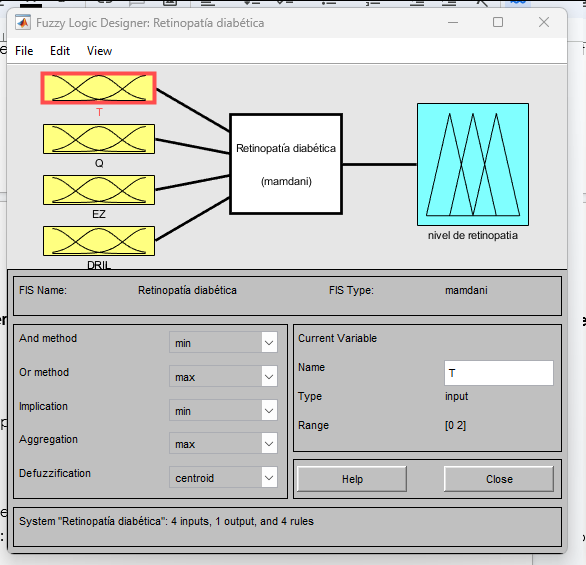
2 3 2 2 4 1 1 % Atrófica

];

Luego, se guarda el sistema difuso en la variable ‘a’:

a = addRule(a,ruleList);

Y se observa el sistema por medio de la interfaz gráfica:



Y finalmente se evalúa el sistema con los valores de antecedente y consecuente

Y = evalfis([0 3 0 0], a)

Dando como resultado:

