IMPLEMENTAREA ÎN MEDIUL MATLAB-SIMULINK A UNOR STRATEGII DE MODELARE BAZATE PE LOGICA FUZZY (1)

MULȚIMI FUZZY. LOGICĂ FUZZY.

1. Consideraţii generale, motivaţie şi obiectiv

Un tip incipient de logică fuzzy a apărut încă din 1920, propus de matematicianul polonez Jan Łukasiewicz (inventatorul notației poloneze). Sistemul său permitea extinderea valorii de adevăr a unei propoziții la toate numerele reale din intervalul [0,1]. Un număr din acest interval era interpretat drept *posibilitatea* ca propoziția considerată să fie adevărată sau falsă. Aceste cercetări au dus la apariția *teoriei posibilității*, o tehnică de raționament în condiții de inexactitate.

În 1965, Lotfi Zadeh a extins teoria posibilității într-un sistem formal de logică matematică. De asemenea, a adus în discuție modalitățile de lucru cu termeni nuanțați ai limbajului natural. Aceast instrument de reprezentare si manipulare a termenilor nuanțați se numeste **logica fuzzy**. Logica tradițională consideră că un obiect poate aparține sau nu unei mulțimi. Logica fuzzy permite o interpretare mai flexibilă a noțiunii de apartenență. Astfel, mai multe obiecte pot aparține unei mulțimi în grade diferite.

Prin parcurgerea acestei ședințe de laborator, studentul va acumula cunoștințele fundamentale privitoare la conceptul de mulțime fuzzy și a operațiilor cu mulțimi fuzzy prin rezolvarea analitică de probleme și prin intermediul simulării în MATLAB deosebit de utile pentru consolidarea abordărilor teoretice generale.

2. Breviar teoretic

Logica fuzzy poate fi studiată plecând de la conceptul de mulţime fuzzy. O mulţime fuzzy este un set neclar, dar foarte bine delimitat. Poate conține elemente cu un anumit grad de apartenețǎ la grupul respectiv.

Pentru a înțelege mai bine noțiunea de mulţime fuzzy, se consideră mai întâi definiția clasică a mulţimilor. O mulţime în sens clasic (denumită în continuare şi *mulţime crisp*) este o colecţie de obiecte bine definte, distincte, dintr-un anumit spaţiu (univers). Faptul că un element oarecare al spaţiului aparţine sau nu unei anumite mulţimi crisp poate fi modelat matematic prin existenţa unei funcţii binare care indică apartenenţa elementelor spaţiului la mulţimea considerată. Această funcţie ia valoarea 1 dacă un element aparţine mulţimii şi valoarea 0 în caz contrar.

2.1. Mulţimi fuzzy

2.1.1. Definiţia unei mulţimi fuzzy

O mulţime fuzzy este extensia unui set clasic. Dacă considerăm *X universul de discurs* și notăm elementele sale cu *x*, atunci un set fuzzy *A* din *X* este definit ca mulțimea perechilor ordonate

$$A=\{(x,\mu_A(x))\,|\,x\in X\},$$

unde $\mu_A : X \to [0,1]$ se numește *funcție de apartenență*. Această funcție asociază fiecărui element *x* din *X* o valoare $\mu_A(x)$ cuprinsă între 0 și 1 reprezentând *gradul de apartenență* al acestuia la mulţimea fuzzy *A*.

Construcţia unei mulţimi fuzzy presupune: identificarea universului de discurs şi specificarea unei funcţii de apartenenţă adecvate (are caracter subiectiv, depinde de persoana care face aprecieri). O mulţime fuzzy este în mod unic definită prin funcţia sa de apartenenţă.

În cazul în care *X* este o colecţie discretă de obiecte, o mulţime fuzzy definită pe *X* poate fi specificată printr-o relaţie de forma $A=\sum_{x_i\in X}\mu_A(x_i)/x_i$. Dacă X este spaţiu continuu atunci se poate utiliza reprezentarea $A=\int_X \mu_A(x)/x$. În aceste relaţii, simbolurile $\sum$ şi $\int$ precizează de fapt reuniunea tuturor perechilor ordonate $(x,\mu_A(x))$ nu indică adunarea sau integrarea. De asemenea, simbolul $/$ este doar un marker, nu are semnificaţia de împărţire.

2.1.2. Operaţii cu mulţimi fuzzy

Operaţiile cu mulţimi fuzzy reprezintă extensii naturale ale operaţiilor cu mulţimi crisp. În continuare considerăm două mulţimi fuzzy *A* şi *B* definite pe acelaşi univers de discurs *X*,

$$A=\{(x,\mu_A(x))\,|\,x\in X\} \text{ şi } B=\{(x,\mu_B(x))\,|\,x\in X\}.$$

Mulţimile fuzzy *A* şi *B* sunt ***egale*** dacă funcţiile lor de apartenenţă coincid, adică $\mu_A(x)=\mu_B(x)$ pentru orice $x\in X$.

Mulţimea A este ***conţinută*** în mulţimea B (sau *A* este *submulţime* a lui *B*), notată $A\subseteq B$, dacă $\mu_A(x)\le\mu_B(x)$, pentru orice $x\in X$. În cazul în care inegalitatea precedentă este strictă, $\mu_A(x)<\mu_B(x)$, pentru orice $x\in X$, se spune că mulţimea A este ***submulţime proprie*** (strictă) a mulţimii B.

Intersecţia(*conjuncţia*) mulţimilor fuzzy A şi B este mulţimea fuzzy notată $A\cap B$ (sau $A\,\text{AND}\,B$) a cărei funcţie de apartenenţă este:

$$\mu_{A\cap B}(x)=\min\{\mu_A(x),\mu_B(x)\}=\mu_A(x)\wedge\mu_B(x),x\in X$$

Reuniunea(*disjuncţia*) mulţimilor fuzzy A şi B este mulţimea fuzzy notată $A\cup B$ (sau $A\,\text{OR}\,B$) a cărei funcţie de apartenenţă este:

$$\mu_{A\cup B}(x)=\max\{\mu_A(x),\mu_B(x)\}=\mu_A(x)\vee\mu_B(x),x\in X$$

Complementul (*negaţia*) mulţimii fuzzy *A* este mulţimea fuzzy notată $\bar{A}$ (sau $\text{NOT}\,A$) având funcţia de apartenenţă:

$$\mu_{\bar{A}}(x)=1-\mu_A(x)\;\forall x\in X$$

Diferenţa a două mulţimi crisp este definită conform relaţiei $A-B=A\cap\bar{B}$. Pentru mulţimi fuzzy există două moduri de a defini diferenţa:

a) ***diferenţa simplă*** este definită cu ajutorul operaţiilor standard de intersecţie şi complementariere, pe baza relaţiei $A-B=A\cap\bar{B}$, astfel că
$$\mu_{A-B}(x)=\min\{\mu_A(x),1-\mu_B(x)\}$$
b) ***diferenţa mărginită***, notată $A\,!\,B$, este mulţimea fuzzy având funcţia de apartenenţă:
$$\mu_{A\ominus B}(x)=\max\{0,\mu_A(x)-\mu_B(x)\}$$

2.1.3. T-norme şi T-conorme

Intersecţia şi reuniunea a două mulţimi fuzzy *A* şi *B* definite pe acelaşi univers de discurs *X* reprezintă mulţimi fuzzy pe *X* cărora le corespund funcţii de apartenenţă ca pot fi exprimate pe baza valorilor funcţiilor μ_A şi μ_B cu ajutorul unor operatori notaţi T şi, respectiv, S, astfel

$$\mu_{A\cap B}(x)=\mathrm{T}\left(\mu_A(x),\mu_B(x)\right),\ x\in X,$$
$$\mu_{A\cup B}(x)=\mathrm{S}\left(\mu_A(x),\mu_B(x)\right),\ x\in X.$$

Un operator $\mathrm{T}:[0,1]\times[0,1]\to[0,1]$ este o *T-normă* dacă următoarele proprietăţi sunt satisfăcute:

- condiţii pe frontieră: $\mathrm{T}(0,0)=0;\ \mathrm{T}(a,1)=\mathrm{T}(1,a)=a;$
- monotonicitate: $a\le c$ şi $b\le d:\ \mathrm{T}(a,b)\le\mathrm{T}(c,d);$
- comutativitate: $\mathrm{T}(a,b)=\mathrm{T}(b,a);$
- asociativitate: $\mathrm{T}\left(a,\mathrm{T}(b,c)\right)=\mathrm{T}\left(\mathrm{T}(a,b),c\right).$

Câţiva operatori de tip T-normă frecvent utilizaţi sunt prezentaţi în continuare:

(i) minimum $T_{min}(a,b)=a\wedge b=\min\{a,b\}$

(ii) produs algebric $T_{ap}(a,b)=a\cdot b$

(iii) produs mărginit (Lukasiewicz) $T_{bp}(a,b)=a\cup b=\max\{0,a+b-1\}$

(iv) produs drastic $T_{dp}(a,b)=a\cap b=\begin{cases}a,b=1\\ b,a=1\\ 0,a,b<1\end{cases}$

Un operator $\mathrm{S}:[0,1]\times[0,1]\to[0,1]$ este o *T-conormă* (*S-normă*) dacă următoarele proprietăţi sunt satisfăcute:

- condiţii pe frontieră: $\mathrm{S}(1,1)=1;\ \mathrm{S}(a,0)=\mathrm{S}(0,a)=a;$
- monotonicitate: $a\le c$ şi $b\le d:\ \mathrm{S}(a,b)\le\mathrm{S}(c,d);$
- comutativitate: $\mathrm{S}(a,b)=\mathrm{S}(b,a);$
- asociativitate: $\mathrm{S}\left(a,\mathrm{S}(b,c)\right)=\mathrm{S}\left(\mathrm{S}(a,b),c\right).$

Câţiva operatori de tip T-conormă frecvent utilizaţi sunt prezentaţi în continuare:

(i) maximum $S_{max}(a,b)=a\vee b=\max\{a,b\}$

(ii) sumă algebrică $S_{as}(a,b)=a\hat{+}b=\mathrm{a}+\mathrm{b}-\mathrm{ab}$

(iii) sumă mărginită (Lukasiewicz) $S_{bs}(a,b)=a\oplus b=\min\{1,a+b\}$

(iv) sumă drastică $S_{bs}(a,b)=a\cup b=\begin{cases}a,b=0\\ b,a=0\\ 1,a,b>0\end{cases}$

Proprietăţile impuse operatorilor T-normă şi T-conormă sunt necesare pentru ca aplicând un asemenea pentru două mulţimi crisp să se obţină reuniunea. respectiv intersecţia, uzuală a acestora.

2.2. Implementare în MATLAB a funcţiilor de apartenenţă

Toolboxul ***Fuzzy Logic Toolbox*** din Matlab cuprinde 11 tipuri de funcţii de apatenenţă, construite cu ajutorul unor funcţii elementare:

- Liniară pe porţiuni;
- Distribuţie Gaussiană;
- Curba sigmoidală;
- Curbe polinomiale cuadratice şi cubice.

1) Funcția de apartenență triunghiulară TRIMF

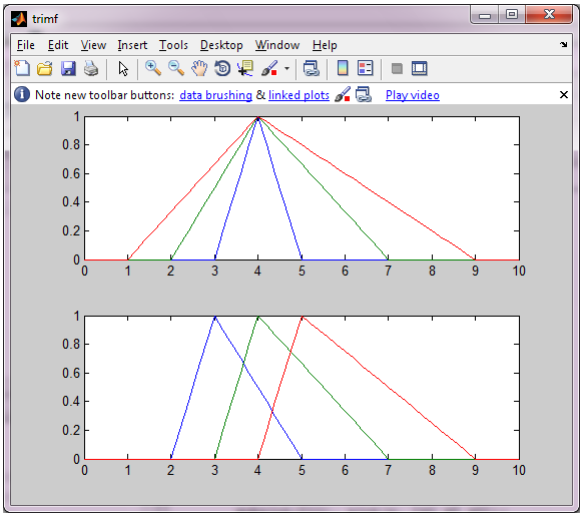
TRIMF(X, PARAMS)

Returnează o matrice reprezentând funcția de apartenență triunghiulară evaluată în X.

PARAMS = [A B C] este un vector cu 3 elemente reprezentând punctele de tăiere ale funcției de apartenență. Se impune ca $A \leq B \leq C$.

Exemplu:

```
x = (0:0.2:10)';
y1 = trimf(x, [3 4 5]);
y2 = trimf(x, [2 4 7]);
y3 = trimf(x, [1 4 9]);
subplot(211), plot(x, [y1 y2 y3]);
y1 = trimf(x, [2 3 5]);
y2 = trimf(x, [3 4 7]);
y3 = trimf(x, [4 5 9]);
subplot(212), plot(x, [y1 y2 y3]);
set(gcf, 'name', 'trimf', 'numbertitle', 'off');
```



2) Funcția de apartenență trapezoidală TRAPMF

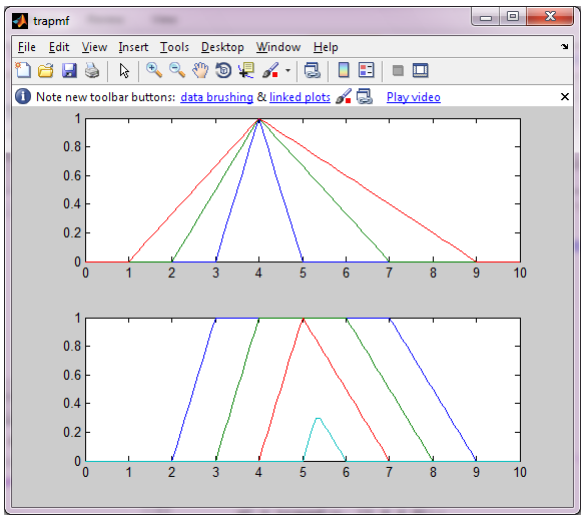
TRAPMF(X, PARAMS)

Returnează o matrice reprezentând funcția de apartenență trapezoidală evaluată în X.

PARAMS = [A B C D] este un vector cu 4 elemente reprezentând punctele de tăiere ale funcției de apartenență. Se impune ca $A \leq B$ *şi* $C \leq D$

Exemplu:

```
x = (0:0.1:10)';
y1 = trapmf(x, [2 3 7 9]);
y2 = trapmf(x, [3 4 6 8]);
y3 = trapmf(x, [4 5 5 7]);
y4 = trapmf(x, [5 6 4 6]);
plot(x, [y1 y2 y3 y4]);
set(gcf, 'name', 'trapmf', 'numbertitle', 'off');
```



3) Funcția de apartenență de tip curbă Gaussiană GAUSSMF GAUSSMF(X, PARAMS)

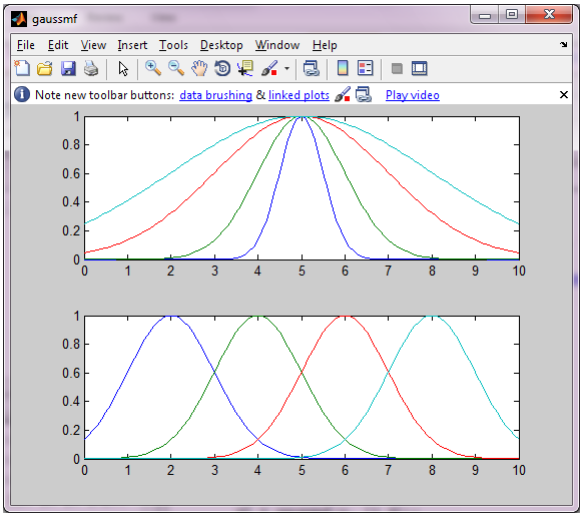
$$GAUSSMF(X, [SIGMA, C]) = e^{-\frac{(X-C)^2}{2 \cdot SIGMA^2}}$$

Returnează o matrice reprezentând funcția de apartenență Gaussiană evaluată în X.

PARAMS = [SIGMA, C] este un vector cu 2 elemente care determină forma și poziția funcției de apartenență

Exemplu:

```
x = (0:0.1:10)';
y1 = gaussmf(x, [0.5 5]);
y2 = gaussmf(x, [1 5]);
y3 = gaussmf(x, [2 5]);
y4 = gaussmf(x, [3 5]);
subplot(211); plot(x, [y1 y2 y3 y4]);
y1 = gaussmf(x, [1 2]);
y2 = gaussmf(x, [1 4]);
y3 = gaussmf(x, [1 6]);
y4 = gaussmf(x, [1 8]);
subplot(212); plot(x, [y1 y2 y3 y4]);
set(gcf, 'name', 'gaussmf', 'numbertitle', 'off');
```



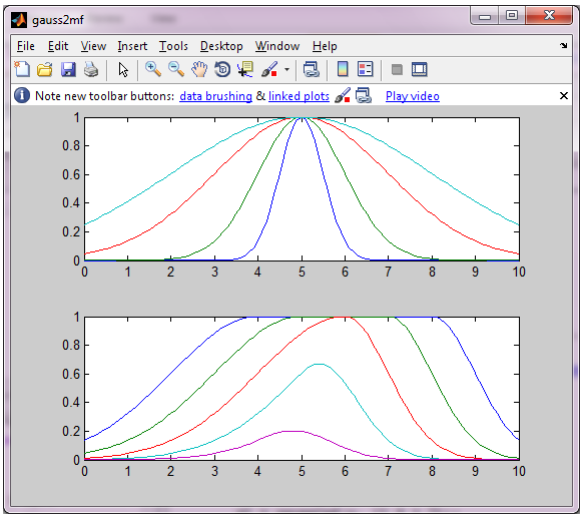
4) Funcția de apartenență obținută prin combinația a două funcții Gaussiene GAUSS2MF GAUSS2MF(X, PARAMS)
PARAMS = [sig1 c1 sig2 c2]

Prima funcție, specificată prin sig1 și c1 determină forma curbei din stânga, iar cea de-a doua determină forma curbei din dreapta.

Când c1<c2, funcția gauss2mf atinge valoarea maximă 1, altfel valoarea sa este subunitară.

Exemplu:

```
x = (0:0.1:10)';
y1 = gauss2mf(x, [2 4 1 8]);
y2 = gauss2mf(x, [2 5 1 7]);
y3 = gauss2mf(x, [2 6 1 6]);
y4 = gauss2mf(x, [2 7 1 5]);
y5 = gauss2mf(x, [2 8 1 4]);
plot(x, [y1 y2 y3 y4 y5]);
set(gcf, 'name', 'gauss2mf', 'numbertitle', 'off');
```



5) Funcția de apartenență tip clopot generalizat GBELLMF GBELLMF(X, PARAMS)

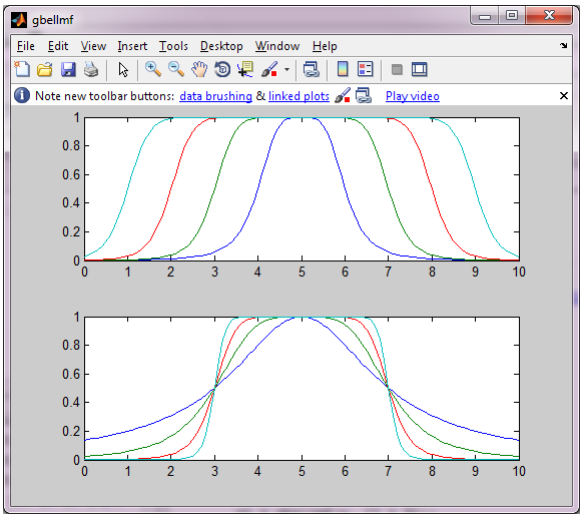
Returnează o matrice reprezentând funcția de apartenență tip clopot generalizat evaluată în X.

PARAMS este un vector cu 3 elemente ce determină forma și poziția funcției de apartenență

$$GBELLMF(X,[A,B,C]) = \frac{1}{\left(1+\left|\frac{X-C}{A}\right|\right)^{2B}}$$

Exemplu:

```
x = (0:0.1:10)';
y1 = gbellmf(x, [1 2 5]);
y2 = gbellmf(x, [2 4 5]);
y3 = gbellmf(x, [3 6 5]);
y4 = gbellmf(x, [4 8 5]);
subplot(211); plot(x, [y1 y2 y3 y4]);
y1 = gbellmf(x, [2 1 5]);
y2 = gbellmf(x, [2 2 5]);
y3 = gbellmf(x, [2 4 5]);
y4 = gbellmf(x, [2 8 5]);
subplot(212); plot(x, [y1 y2 y3 y4]);
set(gcf, 'name', 'gbellmf', 'numbertitle', 'off');
```



6) Funcția de apartenență sigmoidală SIGMF SIGMF(X, PARAMS)

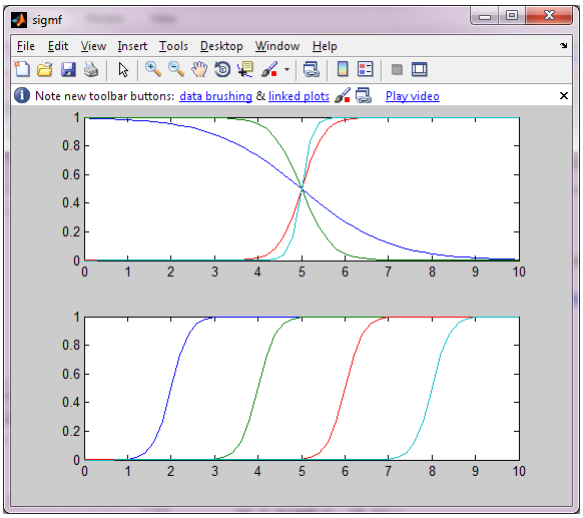
Returnează o matrice reprezentând funcția de apartenență sigmoidală evaluată în X.

PARAMS este un vector cu 2 elemente ce determină forma și poziția funcției de apartenență.

$$SIGMF(X,[A,C]) = \frac{1}{1+e^{-A\cdot(X-C)}}$$

Exemplu:

```
x = (0:0.2:10)';
y1 = sigmf(x, [-1 5]);
y2 = sigmf(x, [-3 5]);
y3 = sigmf(x, [4 5]);
y4 = sigmf(x, [8 5]);
subplot(211); plot(x, [y1 y2 y3 y4]);
y1 = sigmf(x, [5 2]);
y2 = sigmf(x, [5 4]);
y3 = sigmf(x, [5 6]);
y4 = sigmf(x, [5 8]);
subplot(212); plot(x, [y1 y2 y3 y4]);
set(gcf, 'name', 'sigmf', 'numbertitle', 'off');
```

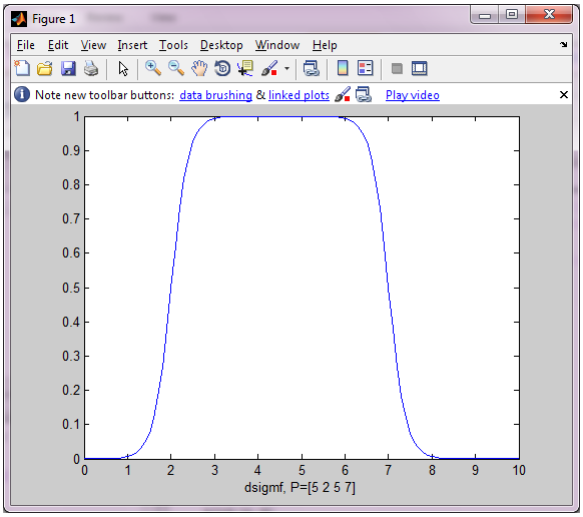


7) **Funcția de apartenență de tip diferență a două funcții sigmoidale DSIGMF** DSIGMF(X,[a1 c1 a2 c2])

Funcția de apartenență DSIGMF depinde de 4 parametri a1, c1, a2 și c2 și este exprimată ca diferența dintre cele 2 funcții sigmoidale f1(x; a1, c1) - f2(x; a2, c2)

Exemplu:

```
x=0:0.1:10;
y=dsigmf(x,[5 2 5 7]);
plot(x,y);
xlabel('dsigmf, P=[5 2 5 7]');
```



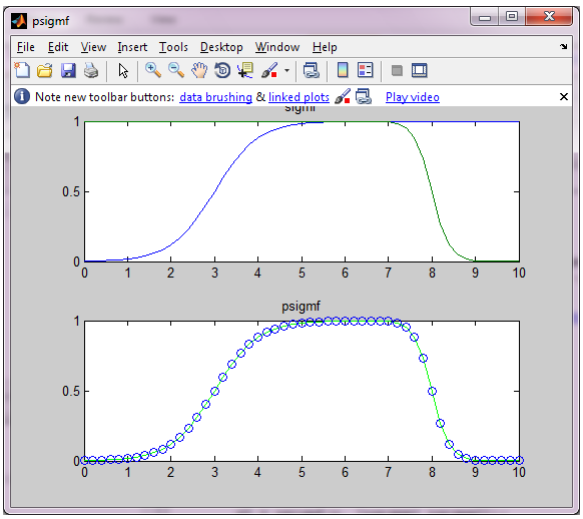
8) **Funcția de apartenență de tip produs a două funcții sigmoidale PSIGMF** PSIGMF(X, PARAMS)

Returnează o matrice reprezentând produsul celor două funcții sigmoidale evaluate în X.

PSIGMF(X, PARAMS) = SIGMF(X, PARAMS(1:2)).*SIGMF(X, PARAMS(3:4));

Exemplu:

```
x = (0:0.2:10)';
params1 = [2 3];
y1 = sigmf(x, params1);
params2 = [-5 8];
y2 = sigmf(x, params2);
y3 = psigmf(x, [params1 params2]);
subplot(211);
plot(x, y1, x, y2); title('sigmf');
subplot(212);
plot(x, y3, 'g-', x, y3, 'o'); title('psigmf');
set(gcf, 'name', 'psigmf', 'numbertitle', 'off');
```



9) Funcția de apartenență s SMF

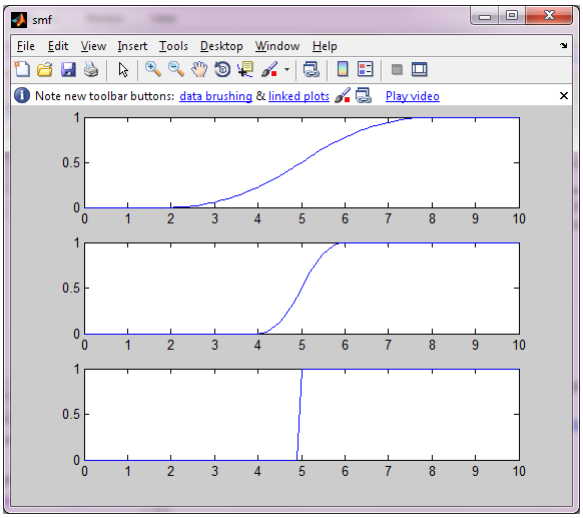
Returnează o matrice reprezentând funcția de apartenență de tip s evaluată în X.

SMF(X, PARAMS)

PARAMS = [X0 X1] este un vector cu 2 elemente ce determină punctele de tăiere ale funcției de apartenență.

Exemplu:

```
x = 0:0.1:10;
subplot(311); plot(x, smf(x, [2 8]));
subplot(312); plot(x, smf(x, [4 6]));
subplot(313); plot(x, smf(x, [6 4]));
set(gcf, 'name', 'smf', 'numbertitle', 'off');
```



10) Funcția de apartenență z ZMF

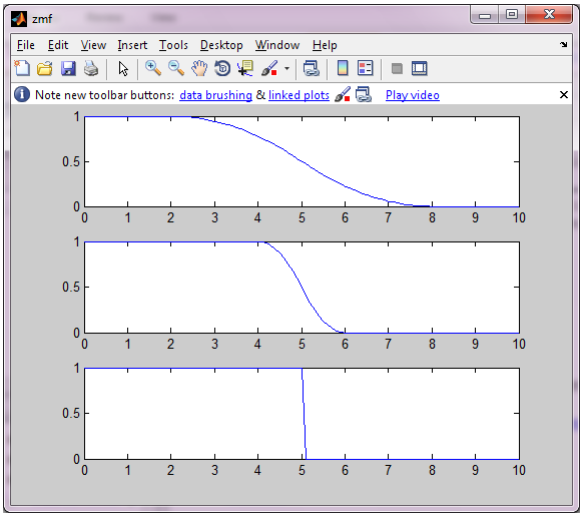
Returnează o matrice reprezentând funcția de apartenență de tip z evaluată în X.

ZMF(X, PARAMS)

PARAMS = [X1 X0] este un vector cu 2 elemente ce determină punctele de tăiere ale funcției de apartenență.

Exemplu:

```
x = 0:0.1:10;
subplot(311); plot(x, zmf(x, [2 8]));
subplot(312); plot(x, zmf(x, [4 6]));
subplot(313); plot(x, zmf(x, [6 4]));
set(gcf, 'name', 'zmf', 'numbertitle', 'off');
```

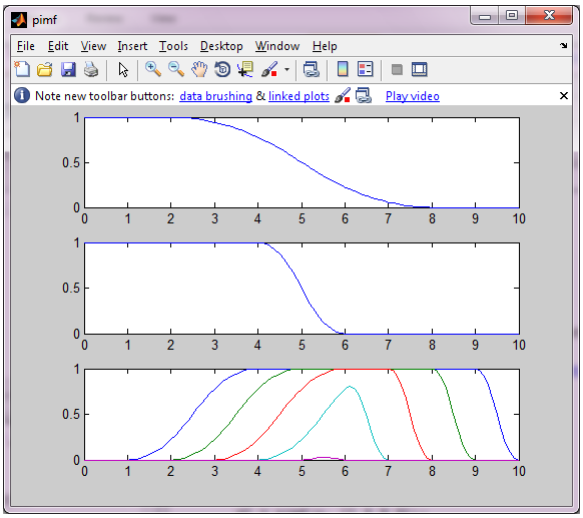


11) Funcţia de apartenenţă pi PIMF

PIMF(X, PARAMS) = SMF(X, PARAMS(1:2)).*ZMF(X, PARAMS(3:4))

Exemplu:

```
x = (0:0.1:10)';
y1 = pimf(x, [1 4 9 10]);
y2 = pimf(x, [2 5 8 9]);
y3 = pimf(x, [3 6 7 8]);
y4 = pimf(x, [4 7 6 7]);
y5 = pimf(x, [5 8 5 6]);
plot(x, [y1 y2 y3 y4 y5]);
set(gcf, 'name', 'pimf', 'numbertitle', 'off');
```



2.3. Logica fuzzy

Noţiunile referitoare la mulţimi fuzzy pot fi extinse pentru a opera cu **relaţii fuzzy** şi, în final, pentru a efectua „judecăţi” bazate pe logica fuzzy. În acest sens, un principiu de bază în teoria mulţimilor fuzzy este *principiul extensiei* (eng. *extension principle*) care furnizează o procedură de extindere a domeniului unei expresii matematice de la mulţimi crisp la mulţimi fuzzy. Pentru a ilustra aplicarea acestui principiu, considerăm o funcţie $f: X \to Y$ şi o mulţime fuzzy A definită pe X prin:

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \ldots + \frac{\mu_A(x_n)}{x_n}.$$

Dacă aplicaţia f este injectivă, imaginea mulţimii A prin f este mulţimea fuzzy:

$$B = f(A) = \frac{\mu_A(x_1)}{y_1} + \frac{\mu_A(x_2)}{y_2} + \ldots + \frac{\mu_A(x_n)}{y_n},$$

unde $y_i = f(x_i)$, $i = 1,2,\ldots n$.

Dacă aplicaţia f nu este injectivă, atunci există $x_1, x_2 \in X$, $x_1 \neq x_2$, astfel încât $f(x_1) = f(x_2) = y^*$, $y^* \in Y$. În acest caz, gradul de apartenenţă a elementului y^* la B este dat de maximul dintre valorile $\mu_A(x_1)$ şi $\mu_A(x_2)$, deoarece valoarea y^* poate fi obţinută aplicând funcţia f lui x_1 sau lui x_2.

Generalizând această observaţie, imaginea mulţimii fuzzy $A = \{(x, \mu_A(x)) \mid x \in X\}$, prin aplicaţia $f: X \to Y$ este mulţimea fuzzy $B = \{(y, \mu_B(y)) \mid y \in Y\}$, a cărei funcţie de apartenenţă este

$$\mu_B(y) = \begin{cases} \sup_{x \in f^{-1}(y)} \{\mu_A(x)\}, & \text{dacă } f^{-1}(y) \neq \varnothing, \\ 0, & \text{dacă } f^{-1}(y) = \varnothing. \end{cases}$$

În cazul în care $f: X_1 \times X_2 \times \ldots \times X_n \to Y$ este o funcţie multivariabilă şi A_i este mulţime fuzzy în universul X_i, pentru $i=\overline{1,n}$, putem defini mulţimea fuzzy $B=\{(y, \mu_B(y)) / y \in Y\}$ având funcţia de apartenenţă

$$\mu_B(y)=\begin{cases} \sup\limits_{x \in f^{-1}(y)} \min\{\mu_{A_1}(x_1), \mu_{A_2}(x_2), \ldots, \mu_{A_n}(x_n)\}, & \text{dacă } f^{-1}(y) \neq \varnothing, \\ 0, & \text{dacă } f^{-1}(y)=\varnothing. \end{cases}$$

Fie X şi Y două universuri de discurs. O relaţie fuzzy R între elementele din X şi cele din Y este o mulţime fuzzy definită prin funcţia de apartenenţă $\mu_{\mathrm{R}}: X \times Y \to [0,1]$ (unde $X \times Y$ notează produsul cartezian al mulţimilor crisp *X* şi *Y*):

$$\mathrm{R}=\{((x,y), \mu_{\mathrm{R}}(x,y)) \mid (x,y) \in X \times Y\}.$$

Numărul $\mu_{\mathrm{R}}(x,y) \in [0,1]$ caracterizează „tăria” legăturii dintre elementele x şi y.

În cazul în care spaţiile X şi Y au număr finit de elemente, $X=\{x_1, x_2, \ldots, x_n\}$, $Y=\{y_1, y_2, \ldots, y_m\}$, o relaţie binară R pe $X \times Y$ poate fi reprezentată printr-o matrice $\boldsymbol{R}=[r_{ij}] \in [0,1]^{n \times m}$, având ca elemente valorile funcţiei de apartenenţă μ_{R}:

$$r_{ij}=\mu_{\mathrm{R}}(x_i, y_i),\ i=\overline{1,n},\ j=\overline{1,m}.$$

În cazul în care $Y=X$, despre o relaţie fuzzy R de la X la X se spune că este:

- *reflexivă*: dacă $\mu_{\mathrm{R}}(x,x)=1$ şi $\mu_{\mathrm{R}}(x,y)<1$, pentru $x \neq y$;
- *antireflexivă*: dacă $\mu_{\mathrm{R}}(x,x)=0$ şi $\mu_{\Re}(x,y)<1$, pentru $x \neq y$;
- *simetrică*: dacă $\mu_{\mathrm{R}}(x,y)=\mu_{\mathrm{R}}(y,x)$, pentru orice $x,y \in X$;
- *antisimetrică*: dacă $\mu_{\mathrm{R}}(x,y) \neq \mu_{\mathrm{R}}(y,x)$, pentru $x \neq y$.

Privind relaţiile fuzzy ca mulţimi fuzzy definite pe produsul cartezian al universurilor de discurs, relaţiilor le pot fi aplicate toate **operaţiile** specifice mulţimilor. De exemplu, pentru două relaţii fuzzy definite de la *X* la *Y* au sens următoarele noţiuni:

1. *incluziunea*: $\mathrm{R}_1 \subseteq \mathrm{R}_2$ dacă $\mu_{\mathrm{R}_1}(x,y) \leq \mu_{\mathrm{R}_2}(x,y), \forall x \in X, \forall y \in Y$;
2. *intersecţia*, notată $\mathrm{R}_1 \cap \mathrm{R}_2$, pentru care:
$$\mu_{\mathrm{R}_1 \cap \mathrm{R}_2}(x,y)=\min\{\mu_{\mathrm{R}_1}(x,y), \mu_{\mathrm{R}_2}(x,y)\}.$$
3. *reuniunea*, notată $\mathrm{R}_1 \cup \mathrm{R}_2$, pentru care:
$$\mu_{\mathrm{R}_1 \cup \mathrm{R}_2}(x,y)=\max\{\mu_{\mathrm{R}_1}(x,y), \mu_{\mathrm{R}_2}(x,y)\}.$$
4. *complementul*, notat $\bar{\mathrm{R}}$, definit prin:
$$\mu_{\bar{\mathrm{R}}}(x,y)=1-\mu_{\mathrm{R}}(x,y).$$

Alte operaţii specifice relaţiilor fuzzy sunt prezentate în continuare.

Având dată o relaţie binară fuzzy R de la X la Y căreia ii corespunde funcţia de apartenenţă $\mu_{\mathrm{R}}: X \times Y \to [0,1]$, ***proiecţia*** relaţiei R pe universul de discurs X este relaţia unară R_1 definită pe X prin

$$\mu_{\mathrm{R}_1} : X \to [0,1],\ \mu_{\mathrm{R}_1}(x) = \max_{y \in Y} \mu_{\mathrm{R}}(x,y),$$

iar *proiecţia* relaţiei R pe universul de discurs Y este relaţia unară R_2 definită pe Y prin

$$\mu_{\mathrm{R}_2} : Y \to [0,1],\ \mu_{\mathrm{R}_2}(y) = \max_{x \in X} \mu_{\mathrm{R}}(x,y).$$

Invers, având dată o relaţie unară R pe X şi un spaţiu oarecare *Y*, ***extensia cilindrică*** a relaţiei R pe $X \times Y$ este relaţia binară R_C pentru care

$$\mu_{\mathrm{R}_C} : X \times Y \to [0,1],\ \mu_{\mathrm{R}_C}(x,y) = \mu_{\mathrm{R}}(x).$$

Două relaţii fuzzy binare R_1 şi R_2 se pot **compune** numai dacă al doilea spaţiu de definiţie al lui R_1 coincide cu primul spaţiu de definiţie al lui R_2. Astfel, având date două relaţii fuzzy R_1 şi R_2 definite pe $X \times Y$, respectiv pe $Y \times Z$, prin compunerea lor se obţine o relaţie definită pe $X \times Z$.

Există mai multe modalităţi de a defini *compunerea* a două relaţii fuzzy. Cea mai cunoscută dintre acestea este cu **operatorul *max-min*** introdus de Zadeh şi notat $\mathrm{R}_1 \circ \mathrm{R}_2$: pentru care

$$\mu_{\mathrm{R}_1 \circ \mathrm{R}_2} : X \times Z \to [0,1],\ \mu_{\mathrm{R}_1 \circ \mathrm{R}_2}(x,z) = \max_{y \in Y} \left\{ \min \left\{ \mu_{\mathrm{R}_1}(x,y), \mu_{\mathrm{R}_2}(y,z) \right\} \right\}.$$

Dacă spaţiile *X*, *Y* şi *Z* au număr finit de elemente iar relaţiile R_1 şi R_2 sunt prezentate sub forma matricelor relaţionale $\boldsymbol{R}_1$ şi $\boldsymbol{R}_2$, calculul matricei corespunzătoare relaţiei $\mathrm{R}_1 \circ \mathrm{R}_2$ se realizează similar înmulţirii matricelor, operaţiile algebrice uzuale de adunare $(+)$ şi înmulţire $(\cdot)$ fiind înlocuite cu operatorii max şi, respectiv, min. Din acest motiv compunerea a doua relaţii după această regulă mai poartă denumirea de ***produs max-min.***

Dacă R_1, R_2 şi R_3 sunt relaţii binare pentru care are sens compunerea şi se utilizează produsul max-min al relaţiilor, atunci sunt satisfăcute următoarele proprietăţi:

- *asociativitatea*: $\mathrm{R}_1 \circ (\mathrm{R}_2 \circ \mathrm{R}_3) = (\mathrm{R}_1 \circ \mathrm{R}_2) \circ \mathrm{R}_3$;
- *distributivitatea faţa de reuniune*: $\mathrm{R}_1 \circ (\mathrm{R}_2 \cup \mathrm{R}_3) = (\mathrm{R}_1 \circ \mathrm{R}_2) \cup (\mathrm{R}_1 \circ \mathrm{R}_3)$;
- *distributivitatea faţa de intersecţie*: $\mathrm{R}_1 \circ (\mathrm{R}_2 \cap \mathrm{R}_3) \subseteq (\mathrm{R}_1 \circ \mathrm{R}_2) \cap (\mathrm{R}_1 \circ \mathrm{R}_3)$;
- *monotonicitatea*: dacă $\mathrm{R}_1 \subseteq \mathrm{R}_2$ atunci $\mathrm{R}_1 \circ \mathrm{R}_3 \subseteq \mathrm{R}_2 \circ \mathrm{R}_3$.

Pentru compunerea a două relaţii fuzzy R_1 şi R_2 pe $X \times Y$, respectiv pe $Y \times Z$, sunt frecvent utilizaţi următorii operatori:

- **operatorul *max-product***, notat $\mathrm{R}_1 * \mathrm{R}_2$, pentru care

$$\mu_{\mathrm{R}_1 * \mathrm{R}_2} : X \times Z \to [0,1],\ \mu_{\mathrm{R}_1 * \mathrm{R}_2}(x,z) = \max_{y \in Y} \left\{ \mu_{\mathrm{R}_1}(x,y) \cdot \mu_{\mathrm{R}_2}(y,z) \right\};$$

- **operatorul *max-average***, notat $\mathrm{R}_1 \oplus \mathrm{R}_2$, pentru care

$$\mu_{\mathrm{R}_1 \oplus \mathrm{R}_2} : X \times Z \to [0,1],\ \mu_{\mathrm{R}_1 \oplus \mathrm{R}_2}(x,z) = \frac{1}{2} \max_{y \in Y} \left\{ \mu_{\mathrm{R}_1}(x,y) + \mu_{\mathrm{R}_2}(y,z) \right\}.$$

Atunci când se doreşte modelarea unui sistem trebuie puse în evidenţă variabilele (conceptele) cu care se lucrează şi valorile pe care le pot lua acestea. Dacă conceptele respective sunt cuantificate într-un sistem fuzzy ele sunt denumite ***variabile lingvistice***, iar caracteristicile corespunzătoare lor reprezintă *termeni lingvistici*.

Formal, o *variabilă lingvistică* este caracterizată printr-un 4-tuplu: $\langle \mathcal{X}, T(\mathcal{X}), X, \mu \rangle$ în care $\mathcal{X}$ reprezintă numele variabilei (eticheta acesteia); $T(\mathcal{X})$ este mulţimea termenilor lingvistici care pot reprezenta valori ale variabilei $\mathcal{X}$; *X* este mulţimea în care termenii lingvistici pot lua valori (universul de discurs care defineşte caracteristice variabilei). Fiecare valoare lingvistică *A* (termen lingvistic) din

$T(\mathcal{X})$ reprezintă o mulţime fuzzy definită pe *X* pentru care este cunoscută funcţia de apartenenţă $\mu_A : X \to [0,1]$.

3. Probleme propuse

Problema 1.

Fie universul de discurs $X = \{1, 2, 3, 4\}$ și mulțimile
$A = \{(1, 0.2), (2, 0.7), (3, 1), (4, 0)\}$ şi $B = \{(1, 0.5), (2, 0.3), (3, 1), (4, 0.1)\}$.
Să se calculeze:

i) $A \cup B$
ii) $A \cap B$
iii) $\bar{A}$ şi $\bar{B}$
iv) $A - B$
v) $A \ominus B$

Problema 2.

Fie X = (0:0.1:10)'. Să se reprezinte grafic următoarele funcții de apartenență:

i) $f_{tri}(X; 2,6,8)$
ii) $f_{trap}(X; 1,3,7,8)$
iii) $f_{gauss}(X; 2,5)$
iv) $f_{gbell}(X; 2,3,5)$
v) $f_{gauss}(X; 3,4,1,8)$
$f_{gauss}(X; 3,6,1,6)$
vi) $f_{sig}(X; 1,0)$
$f_{sig}(X; -1,0)$
vii) $f_{dsig}(X; 2, -5,5,0)$
viii) $f_{psig}(X; 2, -5, -1,0)$
ix) $f_s(X; -5,5)$
x) $f_z(X; -5,5)$
xi) $f_{pi}(X; -7,2,5,9)$

Problema 3.

Se consideră mulțimile fuzzy:

$$A = \left\{\frac{1}{2} + \frac{0.5}{3} + \frac{0.3}{4} + \frac{0.2}{5}\right\}; B = \left\{\frac{0.5}{2} + \frac{0.7}{3} + \frac{0.2}{4} + \frac{0.4}{5}\right\}.$$

Să se calculeze $A \cup B, A \cap B, \bar{A}$ şi $\bar{B}$.

Problema 4.

Se consideră următoarele mulțimi fuzzy:

$$X = \left\{\frac{0.2}{10} + \frac{0.5}{20} + \frac{0.8}{40} + \frac{1.0}{60} + \frac{0.6}{80} + \frac{0.1}{100}\right\}$$
$$Y = \left\{\frac{0.3}{0.5} + \frac{0.6}{1} + \frac{0.9}{1.5} + \frac{1.0}{4} + \frac{0.6}{8} + \frac{0.3}{20}\right\}$$
$$Z = \left\{\frac{0.3}{10} + \frac{0.6}{20} + \frac{0.7}{40} + \frac{0.9}{60} + \frac{1}{80} + \frac{0.5}{100}\right\}$$

i) Calculați produsul cartezian $R = X \times Y$;
ii) Calculați $S_1 = Z \circ R$ și $S_2 = Z * R$ folosind: 1)compunerea max-min; respectiv 2)compunerea max-prod.

Indicații

$\mu_{R_1 \circ R_2}(x,z) = \max_{y \in Y}\{min\{\mu_{R_1}(x,y), \mu_{R_2}(y,z)\}\}$

$\mu_{R_1 * R_2}(x,z) = \max_{y \in Y}\{\mu_{R_1}(x,y) \cdot \mu_{R_2}(y,z)\}.$

```
% Compunerea max-prod
R = input('Introduceti matricea R:');
S = input('Introduceti matricea S:');
[m, n] = size(R);
[a, b] = size(S);
if(n == a)
    for i = 1:m
        for j = 1:b
            c = R(i,:);
            d = S(:,j);
            [f, g] = size(c);
            [h, q] = size(d);
            for l = 1:g
                e(1,l) = c(1,l) * d(l,1);
            end
            t(i,j) = max(e);
        end
    end
    fprintf('\nMatricea compusa prin metoda max-prod este: \n');
    t
else
    display('\nCele doua relatii nu se pot compune\n');
end
```

```
% Compunerea max-min
R = input('Introduceti matricea R:');
S = input('Introduceti matricea S:');
[m, n] = size(R);
[a, b] = size(S);
if(n == a)
    for i = 1:m
        for j = 1:b
            c = R(i,:);
            d = S(:,j);
            f=d';
            e = min(c,f);
            h(i,j) = max(e);
        end
    end
    display('Matricea compusa prin metoda max-prod este:');
    display(h);
else
    display('\nCele doua relatii nu se pot compune\n');
end
```

Problema 5.

Să se calculeze $R = R_1 \oplus R_2$ pentru relațiile fuzzy R_1 și R_2 date prin matricile relaționale:

$$R_1 = \begin{array}{c|ccccc} & y_1 & y_2 & y_3 & y_4 & y_5 \\ x_1 & 0.1 & 0.2 & 0 & 1 & 0.7 \\ x_2 & 0.3 & 0.5 & 0 & 0.2 & 1 \\ x_3 & 0.8 & 0 & 1 & 0.4 & 0.3 \end{array} \qquad R_2 = \begin{array}{c|cccc} & z_1 & z_2 & z_3 & z_4 \\ y_1 & 0.9 & 0 & 0.3 & 0.4 \\ y_2 & 0.2 & 1 & 0.8 & 0 \\ y_3 & 0.8 & 0 & 0.7 & 1 \\ y_4 & 0.4 & 0.2 & 0.3 & 0 \\ y_5 & 0 & 1 & 0 & 0.8 \end{array}$$

Indicație: $R = R_1 \oplus R_2 = \frac{1}{2}\max\{\mu_{R_1}(x,y) + \mu_{R_2}(y,z)\}$

Problema 6.

Fie matricele relaţionale

$$R=\begin{bmatrix}1&1&0&0&0\\1&1&0&0&1\\0&0&1&0&0\\0&0&0&1&0\\0&1&0&0&1\end{bmatrix},\ P=\begin{bmatrix}1&0.5&0.3&0.6&0\\0.5&1&0.7&0.5&0.9\\0.3&0.7&1&0.6&0\\0.6&0.5&0.6&1&0.5\\0&0.9&0&0.5&1\end{bmatrix}\text{ și }Q=\begin{bmatrix}1&0.8&0.4&0.5&0.8\\0.8&1&0.4&0.5&0.9\\0.4&0.4&1&0.4&0.4\\0.5&0.5&0.4&1&0.5\\0.8&0.9&0.4&0.5&1\end{bmatrix}.$$

Scrieți un program Matlab care să verifice dacă:

i) matricea relațională R este reflexivă sau nu;
ii) matricea relațională R este matrice de toleranță sau nu;
iii) matricea relațională P este simetrică sau nu;
iv) matricea relațională P este tranzitivă sau nu;
v) matricea relațională Q este matrice de echivalență sau nu.

Indicații:

```
% Reflexivitatea
r = input('\nIntroduceti matricea R = ');
sum1 = 0;
[m, n] = size(r);
if(m == n)
    for i=1:m
        %to find the reflexivity
        if(r(1,1) == r(i,i) && r(1,1) == 1)
        else
            fprintf('\nRelatia fuzzy nu este reflexiva\n');
            sum1 = 1;
            break;
        end
    end
    if(sum1 ~= 1)
        fprintf('\nRelatia fuzzy este reflexiva\n');
    end
end
```

```
% Simetria
r = input('\nIntroduceti matricea R = ');
sum = 0;
for i = 1:m
    for j = 1:n
        if(r(i,j) == r(j,i))
        else
            fprintf('\nRelatia fuzzy nu este simetrica\n');
            sum = 1;
            break;
        end
    end
    if(sum == 1)
        break;
    end
end
if(sum ~= 1)
    fprintf('\nRelatia fuzzy este simetrica\n');
end
```

```
% Toleranta
r = input('\nIntroduceti matricea R = ');
sum = 0;
sum1 = 0;
[m, n] = size(r);
if(m == n)
    for i = 1:m
        if(r(1,1) == r(i,i) && r(1,1) == 1)
        else
            fprintf('\nRelatia fuzzy nu este reflexiva\n');
            sum1 = 1;
            break;
        end
    end
    if(sum1 ~= 1)
        fprintf('\nRelatia fuzzy este reflexiva si');
    end
```

```
    for i = 1:m
        for j = 1:n
            if(r(i,j) == r(j,i))
            else
                fprintf(' nu este simetrica ');
                sum = 1;
                break;
            end
        end
        if(sum == 1)
            break;
        end
    end
    if(sum ~= 1)
        fprintf(' simetrica ');
    end
end
if(sum1 ~= 1)
    if(sum ~=1)
        fprintf('\n => Relatia fuzzy este o relatie de toleranta\n');
    else
        fprintf('\nRelatia fuzzy nu este o relatie de toleranta\n');
    end
else
    fprintf('\nRelatia fuzzy nu este o relatie de toleranta\n');
end
```

```
% Tranzitivitatea
r = input('\nIntroduceti matricea R = ');
sum2 = 0;
[m, n] = size(r);
for i = 1:m
    for j = 1:n
        for k = n:1
            lambda1 = r(i,j);
            lambda2 = r(j,k);
            lambda3 = r(i,k);
            p = min(lambda1, lambda2);
            if(lambda3 <= p)
                fprintf('\nRelatia fuzzy nu este tranzitiva');
                sum2 = 1;
                break;
            end
        end
        if(sum2 == 1)
            break;
        end
    end
    if(sum2 == 1)
        break;
    end
end
if(sum2 ~= 1)
    fprintf('\nRelatia fuzzy este tranzitiva\n');
end
```

```
% Echivalenta
r = input('\nIntroduceti matricea R = ');
sum = 0;
sum1 = 0;
sum2 = 0;
[m, n] = size(r);
if(m == n)
    for i = 1:m
        if(r(1,1) == r(i,i) && r(1,1) == 1)
        else
            fprintf('\nRelatia fuzzy nu este reflexiva');
            sum1 = 1;
            break;
        end
    end
    if(sum1 ~= 1)
        fprintf('\nRelatia fuzzy este reflexiva');
    end
    for i = 1:m
```

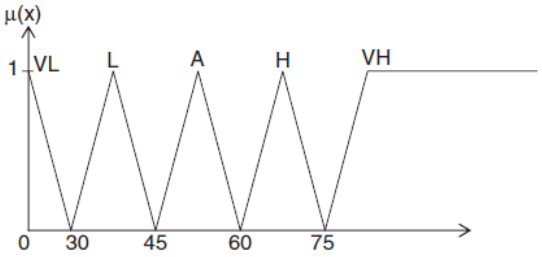
```
        for j = 1:n
            if(r(i,j) == r(j,i))
            else
                fprintf('\nRelatia fuzzy nu este simetrica');
                sum = 1;
                break;
            end
        end
        if(sum == 1)
            break;
        end
    end
    if(sum ~= 1)
        fprintf('\nRelatia fuzzy este simetrica');
    end
end
for i = 1:m
    for j = 1:n
        for k = n:1
            lambda1 = r(i,j);
            lambda2 = r(j,k);
            lambda3 = r(i,k);
            p=min(lambda1, lambda2);
            if(lamda3 <= p)
                fprintf('\nRelatia fuzzy nu este tranzitiva');
                sum2 = 1;
                break;
            end
        end
        if(sum2 == 1)
            break;
        end
    end
    if(sum2==1)
        break;
    end
end
if(sum2 ~= 1)
    fprintf('\nRelatia fuzzy este tranzitiva');
end
if(sum1 ~= 1)
    if(sum ~= 1)
        if(sum2 ~= 1)
            fprintf('\nRelatia fuzzy este o relatie de echivalenta');
        else
            fprintf('\nRelatia fuzzy nu este o relatie de echivalenta');
        end
    else
        fprintf('\nRelatia fuzzy nu este o relatie de echivalenta');
    end
else
    fprintf('\nRelatia fuzzy nu este o relatie de echivalenta');
end
```

Problema 7.

Se consideră universul de discurs al greutății unor persoane măsurată în kg.
Se identifică variabila lingvistică **Greutate** identificată prin:

Foarte ușor (VL)	$w \leq 30$
Ușor (L)	$30 < w \leq 45$
Mediu (A)	$45 < w \leq 60$
Greu (H)	$60 < w \leq 75$
Foarte greu (VH)	$w > 75$

Reprezentarea grafică folosind funcții de apartenență triunghiulare este următoarea:



Folosindu-vă propria intuiție și propriile definiții ale universului de discurs reprezentați grafic funcția de apartenență a următoarelor variabile:

i) Înălțimea coloanei de lichid într-un vas
   a) Foarte plin;
   b) Plin;
   c) Mediu;
   d) Scăzut;
   e) Foarte scăzut.

ii) Vârsta
   a) Foarte tânăr;
   b) Tânăr;
   c) De vârstă mijlocie;
   d) Bătrân;
   e) Foarte bătrân.

Problema 8.

Se consideră controlul vitezei unui motor de curent continuu. Două dintre variabile sunt viteza măsurată în RPM și cuplul T rezultând următoarele două funcții fuzzy de apartenență:

$$S = \left\{\frac{0.2}{x_1} + \frac{0.6}{x_2} + \frac{0.8}{x_3} + \frac{0.6}{x_4} + \frac{0.4}{x_5}\right\}$$

și

$$T = \left\{\frac{0.3}{y_1} + \frac{0.5}{y_2} + \frac{0.6}{y_3} + \frac{1.0}{y_4} + \frac{0.8}{y_5} + \frac{0.3}{y_6} + \frac{0.2}{y_7}\right\}$$

a) Să se găsească relația fuzzy dintre cele două variabile S și T: $R = S \times T$.

b) Se dă și variabila fuzzy I reprezentând curentul indus $I = \begin{matrix} \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{matrix} \begin{bmatrix} z_1 \\ 0.4 \\ 0.5 \\ 0.6 \\ 0.3 \\ 0.7 \\ 0.6 \\ 1.0 \end{bmatrix}$.

Să se gasească $Q = R^{\circ}I$ utilizând compunerea max-min și max-prod.

Problema 9.

Cele trei variabile de interes ale unui tranzistor MOSFET sunt: intensitatea curentului I, tensiunea V și costul C. Se dau următoarele funcții de apartenență pentru tranzistor:

$$I = \left\{\frac{0.4}{0.8} + \frac{0.7}{0.9} + \frac{1}{1} + \frac{0.8}{1.1} + \frac{0.6}{1.2}\right\}$$

$$V = \left\{\frac{0.2}{30} + \frac{0.8}{45} + \frac{1}{60} + \frac{0.9}{75} + \frac{0.7}{90}\right\}$$

$$C = \left\{\frac{0.4}{0.5} + \frac{1}{0.6} + \frac{0.5}{0.7}\right\}$$

a) Să se calculeze produsul cartezian $P = V \times Y$
b) Să se calculeze produsul cartezian $T = I \times C$
c) Folosind compunerea max-min să se calculeze $E = P^{\circ}T$
d) Folosind compunerea max-prod să se calculeze $E = P * T$

Problema 10.

Găsiți relația dintre cele două seturi fuzzy R_1 și R_2 folosind:

a) Compunerea max-min;
b) Compunerea max-prod;
c) Compunerea max-average.

$$R_1 = \begin{array}{c|cccc} & y_1 & y_2 & y_3 & y_4 \\ x_1 & 0.3 & 0.1 & 0.6 & 0.3 \\ x_2 & 0.1 & 1 & 0.2 & 0.1 \end{array} \qquad R_2 = \begin{array}{c|ccc} & z_1 & z_2 & z_3 \\ y_1 & 0.9 & 0.1 & 1 \\ y_2 & 0.1 & 0.5 & 0.4 \\ y_3 & 0.6 & 0.8 & 0.5 \\ y_4 & 0.1 & 0 & 0 \end{array}$$

Problema 11.

Găsiți relația dintre cele două seturi fuzzy utilizînd metoda de compunere max-average:

$$A = \begin{array}{c|ccc} & y_1 & y_2 & y_3 \\ x_1 & 0.1 & 0.5 & 0.6 \\ x_2 & 0.4 & 0.8 & 0.3 \end{array} \qquad B = \begin{array}{c|cc} & z_1 & z_2 \\ y_1 & 0.4 & 0.8 \\ y_2 & 0.6 & 0.5 \\ y_3 & 1 & 0.8 \end{array}$$

Problema 12.

Utilizând compunerea max-min să se determine matricea relațională dintre R_1 și R_2:

$$R_1 = \begin{array}{c|ccccc} & y_1 & y_2 & y_3 & y_4 & y_5 \\ x_1 & 0.1 & 0.2 & 0.1 & 1 & 0.8 \\ x_2 & 0.4 & 0.5 & 0 & 0.2 & 1 \\ x_3 & 0.9 & 0.2 & 0.4 & 0.3 & 0.2 \end{array} \qquad R_2 = \begin{array}{c|cccc} & z_1 & z_2 & z_3 & z_4 \\ y_1 & 0.8 & 0.1 & 0.5 & 0.4 \\ y_2 & 0.3 & 0.9 & 0.8 & 0.1 \\ y_3 & 1 & 0.2 & 0.6 & 0.1 \\ y_4 & 0.4 & 0.2 & 0.3 & 0 \\ y_5 & 0.1 & 1 & 0.8 & 0.7 \end{array}$$

Problema 13.

Să se discute proprietatea de reflexivitate pentru următoarea relație fuzzy:

$$R = \begin{bmatrix} 1 & 0.7 & 0.3 \\ 0.4 & 0.5 & 0.8 \\ 0.7 & 0.5 & 1 \end{bmatrix}$$

Problema 14.

Să se verifice dacă matricea următoare este matricea unei relații de echivalență sau nu:

$$R = \begin{bmatrix} 1 & 0.8 & 0.4 & 0.5 & 0.6 \\ 0.5 & 1 & 0.4 & 0.3 & 0.9 \\ 0.4 & 0.2 & 1 & 0.4 & 0.4 \\ 0.5 & 0.5 & 0.3 & 1 & 0.3 \\ 0.4 & 0.9 & 0.4 & 0.7 & 1 \end{bmatrix}$$