



UNIVERSITATEA BABEŞ-BOLYAI
Facultatea de Matematică și Informatică



INTELIGENȚĂ , ARTIFICIALĂ

Sisteme inteligente

Sisteme bazate pe reguli în medii
incerte

Laura Dioșan

Sumar

A. Scurtă introducere în Inteligența Artificială (IA)

B. Rezolvarea problemelor prin căutare

- Definirea problemelor de căutare
- Strategii de căutare
 - Strategii de căutare neinformate
 - Strategii de căutare informate
 - Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
 - Strategii de căutare adversială

C. Sisteme inteligente

- Sisteme care învață singure
 - Arbori de decizie
 - Rețele neuronale artificiale
 - Mașini cu suport vectorial
 - Algoritmi evolutivi
- Sisteme hibride
- Sisteme bazate pe reguli în medii certe
- Sisteme bazate pe reguli în medii incerte (Bayes, factori de certitudine, Fuzzy)

Materiale de citit și legături utile

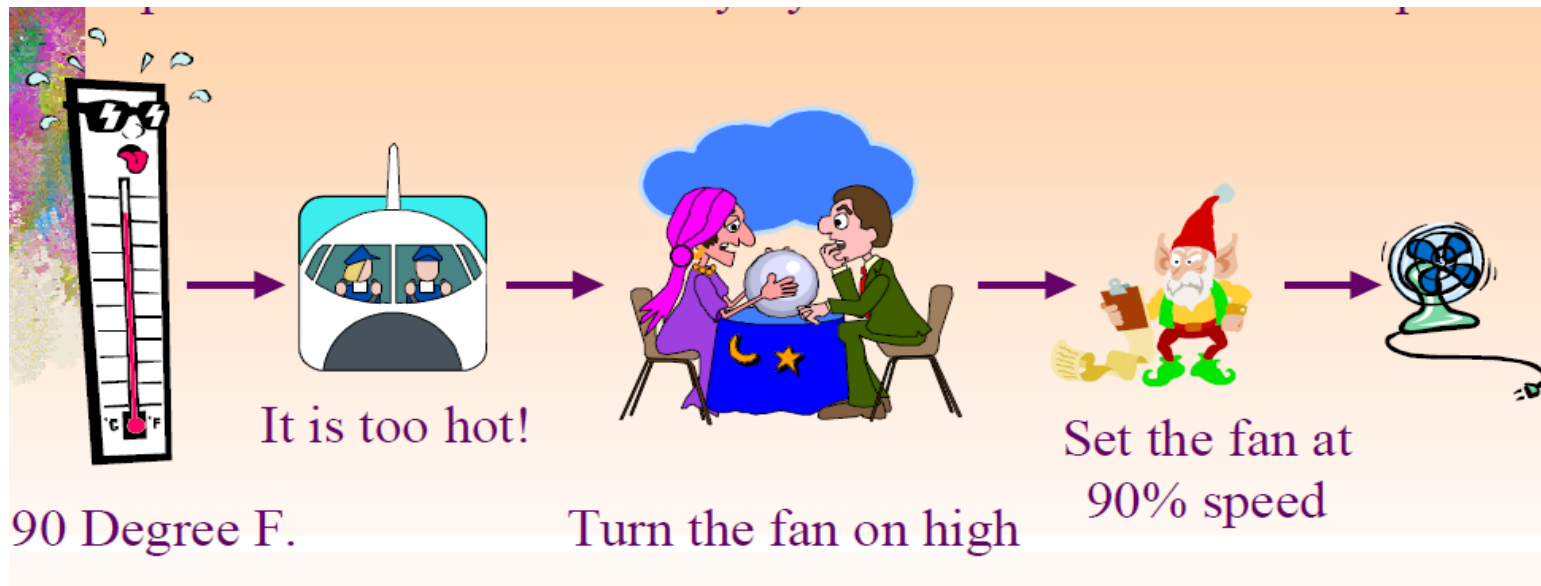
- ❑ capitolul V din *S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995*
- ❑ capitolul 3 din *Adrian A. Hopgood, Intelligent Systems for Engineers and Scientists, CRC Press, 2001*
- ❑ capitolul 8 și 9 din *C. Groșan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011*

Conținut

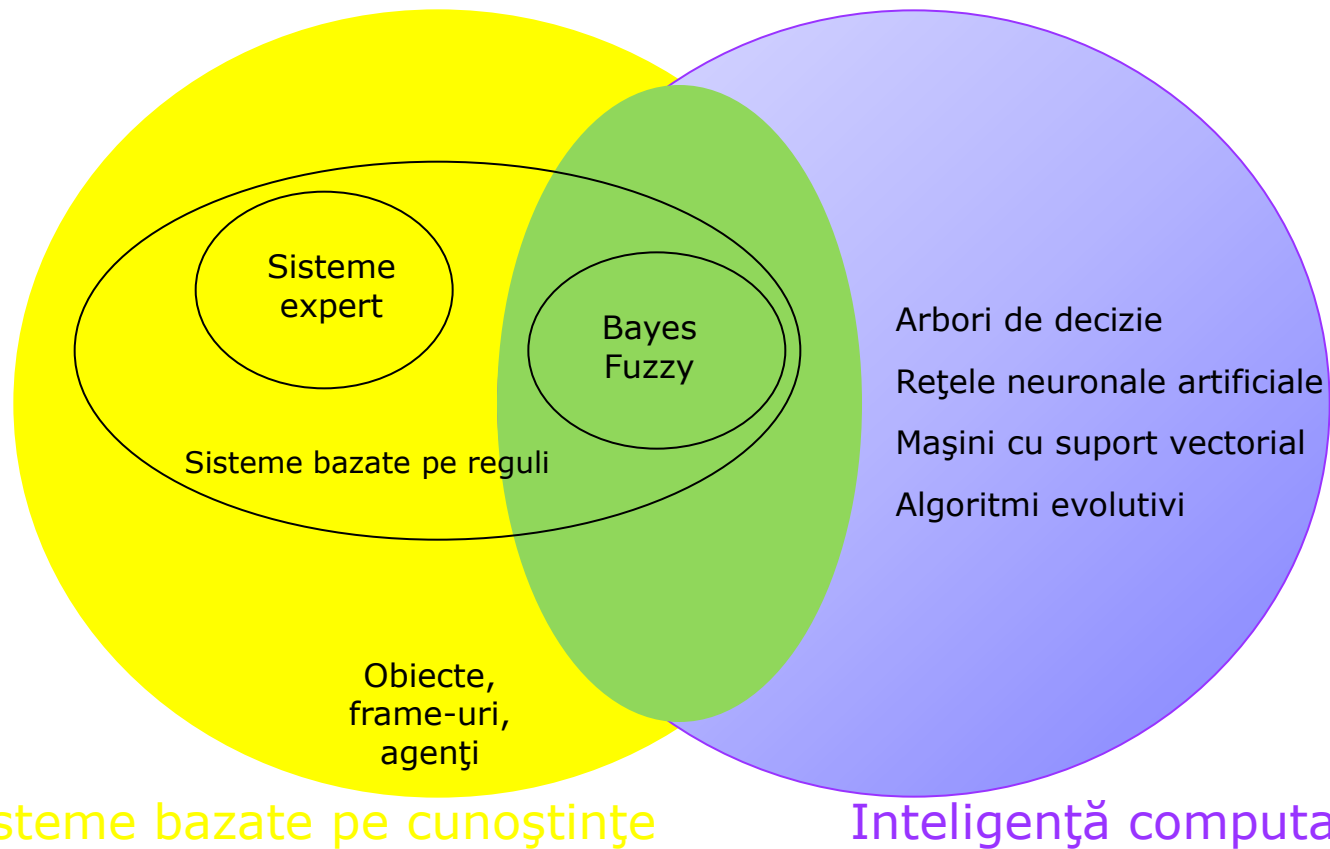
□ Sisteme inteligente

■ Sisteme bazate pe cunoștințe

□ Sisteme bazate pe reguli în medii incerte



Sisteme inteligente



Sisteme inteligente–sisteme bazate pe cunoștințe (SBC)

- sistemele computaționale – alcătuite din 2 module principale (roluri):
 - Domeniul de cunoștințe (baza de cunoștințe – BC – *knowledge base*)
 - Informațiile specifice despre un domeniu
 - Modulul de control (MC – *inference engine*)
 - Regulile prin care se pot obține informații noi
 - Algoritmi independenți de domeniu

Sisteme inteligente – SBC

Baza de cunoștințe (BC)

□ Conținut

- Informații (exprimate într-o anumită reprezentare – ex. propoziții) despre mediu
- informații necesare pentru înțelegerea, formularea și rezolvarea problemelor
- mulțime de propoziții (exprimate/reprezentate într-un limbaj formal) care descriu mediul
 - reprezentare ușor interpretabilă de către calculator → limbaj de reprezentare a cunoștințelor
 - mecanismul de obținere a unor propoziții noi pe baza celor vechi → inferență/raționare

□ Tipologie

- cunoștințe exacte (perfecte)
 - Raționarea exactă se bazează pe logica clasică
 - *IF A is true THEN A is \neg false*
 - *IF B is false THEN B is \neg true*
- cunoștințe imperfecte (nesigure, incerte)
 - Inexacte
 - Incomplete
 - Incomensurabile

Sisteme inteligente – SBC

- Surse ale incertitudinii
 - Imperfecțiunea regulii
 - Nesiguranța evidenței (dovezii)
 - Încrederea în concluzie trebuie scalată
 - Utilizarea unui limbaj vag, imprecis

- Moduri de exprimare a incertitudinii
 - Probabilități
 - Logica fuzzy
 - Teorema lui Bayes
 - Teoria Dempster-Shafer

- Moduri de reprezentare a incertitudinii
 - Cu ajutorul unui singur număr → factor de certitudine, confidență (încredere), valoare de adevăr
 - Cât de siguri suntem că respectivele date sunt valide
 - Cu ajutorul a 2 numere → logica intervalelor
 - Min → limita inferioară a certitudinii (încredere, necesitate)
 - Max → limita superioară a certitudinii (plauzabilitate, posibilitate)

Sisteme inteligente – SBC

- Tehnici de raționare în medii nesigure
 - Teoria Bayesiană – metodă probabilistică
 - Teoria certitudinii
 - Teoria posibilității (logica fuzzy)
- } Metode euristice

Sisteme inteligente – SBC – factori de certitudine

□ sisteme de tip Bayes

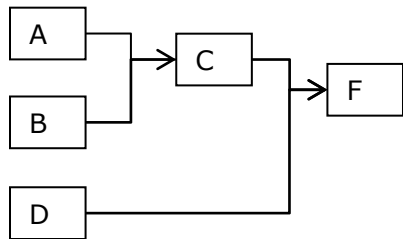
- SBR (Sisteme expert) în care faptele și regulile sunt probabilistice

□ sisteme cu factori de certitudine

- SBR (sisteme expert) în care faptele și regulile au asociate câte un factor de certitudine (FC)/coeficient de încredere
- Un fel de sisteme de tip Bayes în care probabilitățile sunt înlocuite cu factori de certitudine

- Dacă A și B atunci C

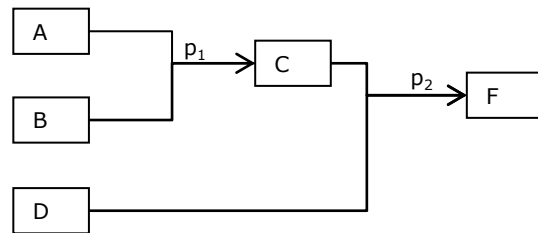
- Dacă C și D atunci F



SBR classic

- Dacă A și B atunci C [cu prob p_1]

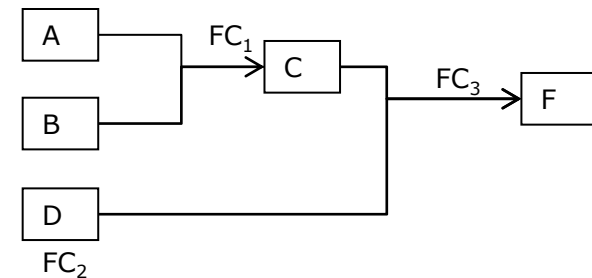
- Dacă C și D atunci F [cu prob p_2]



SBR de tip Bayes

- Dacă A și B atunci C [FC_1]

- Dacă C și D [FC_2] atunci F [FC_3]



SBR cu FC

Sisteme inteligente – SBC – factori de certitudine

□ SBR de tip Bayes vs. SBR cu FC

Bayes	FC
Teorie probabilităților este veche și fundamentată matematic	Teoria FC este nouă și fără demonstrării matematice
Necesită existența unor informații statistice	Nu necesită existența unor date statistice
Propagarea încrederii crește în timp exponențial	Informația circulă repede și eficient în SBR
Este necesar calculul apriori a probabilităților	Nu este necesar calculul apriori a probabilităților
Ipotezele susținute de probe pot să fie independente sau nu	Ipotezele susținute de probe sunt independente

Sisteme inteligente – SBC – sisteme de tip Bayes

- Elemente de teoria probabilităților
- Conținut și arhitectură
- Tipologie
- Tool-uri
- Avantaje și limite

Sisteme inteligente – SBC – sisteme de tip Bayes

Amintim componența unui SBC

- Baza de cunoștințe (BC) → Modalități de reprezentare a cunoștințelor
 - Logica formală (limbaje formale)
 - Definiție
 - Știința principiilor formale de raționament
 - Componente
 - Sintaxă
 - Semantică
 - Metodă de inferență sintactică
 - Tipologie
 - În funcție de numărul valorilor de adevăr:
 - logică duală
 - logică polivalentă
 - În funcție de tipul elementelor de bază:
 - clasică → primitivele = propoziții (predicate)
 - probabilistică → primitivele = variabile aleatoare
 - În funcție de obiectul de lucru:
 - logica propozițională → se lucrează doar cu propoziții declarative, iar obiectele descrise sunt fixe sau unice (Ionică este student)
 - logica predicatelor de ordin I → se lucrează cu propoziții declarative, cu predicate și cuantificări, iar obiectele descrise pot fi unice sau variabile asociate unui obiect unic (Toți studenții sunt prezenți)
 - Reguli
 - Rețele semantice
- Modulul de control (MC – pentru inferență)

Elemente de teoria probabilităților

- Teorii ale probabilităților
- Concepte de bază
 - Teoria clasică și teoria modernă
 - Eveniment
 - Probabilitate simplă
 - Probabilitate condiționată
 - Axiome

Elemente de teoria probabilităților

□ Teorii ale probabilităților

■ Teoria clasică (*a priori*)

- Propusă de Pascal și Fermat în 1654
- Lucrează cu sisteme ideale:
 - toate posibilele evenimente sunt cunoscute
 - toate evenimentele se pot produce cu aceeași probabilitate (sunt uniform distribuite)
- evenimente discrete
- metode combinatoriale
- spațiul rezultatelor posibile este continuu

■ Teoria modernă

- evenimente continue
- metode combinatoriale
- spațiul rezultatelor posibile este cuantificabil

Elemente de teoria probabilităților

□ Concepte de bază

- Considerăm un experiment care poate produce mai multe ieșiri (rezultate)
 - Ex. *Ev1*: Aruncarea unui zar poate produce apariția uneia din cele 6 fețe ale zarului (deci 6 rezultate)
- Eveniment
 - Definiție
 - producerea unui anumit rezultat
 - Ex. *Ev2*: Apariția feței cu nr 3
 - Ex. *Ev3*: Apariția unei fețe cu un nr par (2,4,6)
 - Tipologie
 - Evenimente independente și mutual exclusive
 - Nu se pot produce simultan
 - Ex. *Ev4*: Apariția feței 1 la aruncarea unui zar și *Ev5*: Apariția feței 3 la aruncarea unui zar
 - Dependente
 - Producerea unor evenimente afectează producerea altor evenimente
 - Ex. *Ev6*: Apariția feței 6 la prima aruncare a unui zar și *Ev7*: Apariția unor fețe a căror numere însumate să dea 8 la 2 aruncări succesive ale unui zar
- Mulțimea tuturor rezultatelor = *sample space* al experimentului
 - Ex. pentru *Ev1*: (1,2,3,4,5,6)
- Mulțimea tuturor rezultatelor tuturor evenimentelor posibile = *power set* (mulțimea părților)

Elemente de teoria probabilităților

□ Concepte de bază

■ Probabilitate simplă $p(A)$

- probabilitatea producerii unui eveniment A independent de alte evenimente (B)
- șansa ca acel eveniment să se producă
- proporția cazurilor de producere a evenimentului în mulțimea tuturor cazurilor posibile
- nr cazurilor favorabile / nr cazurilor posibile
- un număr real în $[0,1]$
 - 0 – imposibilitate absolută
 - 1 – posibilitate absolută
- Ex. $P(Ev1) = 1/6$, $P(Ev3) = 3/6$

■ Probabilitate condiționată $p(A|B)$

- probabilitatea producerii unui eveniment A dependentă de producerea altor evenimente (B)
- proporția cazurilor de producere a evenimentului A și a evenimentului B în mulțimea tuturor cazurilor producerii evenimentului B
- probabilitatea comună / probabilitatea lui B

$$p(A|B) = \frac{p(A \cap B)}{p(B)}$$

Elemente de teoria probabilităților

□ Concepte de bază

■ Axiome

- $0 \leq p(E) \leq 1$ pentru orice eveniment E
- $p(\text{Adevărat}) = 1, p(\text{Fals}) = 0$
- $\sum p(E_i) = 1$
- ⁱDacă A și B sunt independente
 - $p(A \cup B) = p(A) + p(B)$
 - $p(A \cap B) = p(A) * p(B)$
- Dacă A și B nu sunt independente
 - Dacă A depinde de B
 - $p(A \cup B) = p(A) + p(B) - p(A \cap B)$
 - $p(A \cap B) = p(A|B) * p(B)$
 - $p(B \cap A) = p(A \cap B)$
 - $p(A|B) = \frac{p(B|A)p(A)}{p(B)}$ (b)
 - Dacă A depinde de B_1, B_2, \dots, B_n (evenimente mutual exclusive)
 - $p(A) = \sum_{i=1}^n p(A | B_i) p(B_i)$ (a)

Elemente de teoria probabilităților

□ Concepte de bază

■ Exemplu

- Dacă A depinde de 2 evenimente mutual exclusive (B și $\neg B$), FC ec.

$$p(A) = \sum_{i=1}^n p(A | B_i) p(B_i) \quad \text{avem:}$$

- $p(B) = p(B|A)p(A) + p(B|\neg A)p(\neg A)$

- Înlocuind pe $p(B)$ în ec. $p(A | B) = \frac{p(B | A)p(A)}{p(B)}$ se obține ec.:

$$p(A | B) = \frac{p(B | A)p(A)}{p(B | A)p(A) + p(B | \neg A)p(\neg A)} \quad (c)$$

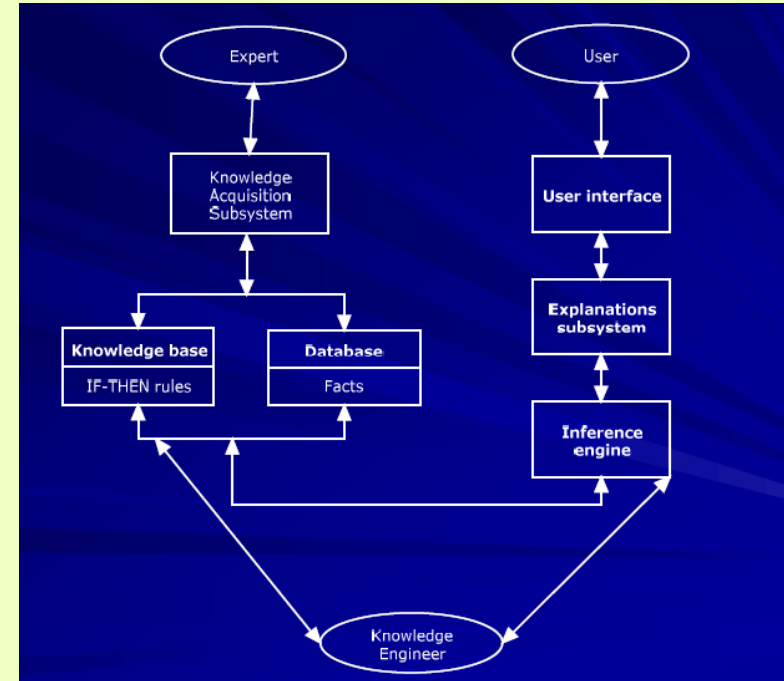
- Ecuația (c) se folosește pentru controlul incertitudinii în sistemele expert

Sisteme inteligente – SBC – sisteme de tip Bayes

Reamintim ca un SBR are următoarea

Arhitectură

- Baza de cunoștințe (BC)
 - Informațiile specifice despre un domeniu
- Modulul de control (MC)
 - Regulile prin care se pot obține informații noi
- Interfața cu utilizatorul
 - permite dialogul cu utilizatorii în timpul sesiunilor de consultare, precum și accesul acestora la faptele și cunoștințele din BC pentru adăugare sau actualizare
- Modulul de îmbogățire a cunoașterii
 - ajută utilizatorul expert să introducă în bază noi cunoștințe într-o formă acceptată de sistem sau să actualizeze baza de cunoștințe.
- Modulul explicativ
 - are rolul de a explica utilizatorilor atât cunoștințele de care dispune sistemul, cât și raționamentele sale pentru obținerea soluțiilor în cadrul sesiunilor de consultare. Explicațiile într-un astfel de sistem, atunci când sunt proiectate corespunzător, îmbunătățesc la rândul lor modul în care utilizatorul percepe și acceptă sistemul



Sisteme inteligente – SBC – sisteme de tip Bayes

Reamintim: SBR – arhitectură

□ baza de cunoștințe

■ Conținut

- Informațiile specifice despre un domeniu sub forma unor
 - fapte – afirmații corecte
 - reguli - euristici speciale care generează informații (cunoștințe)

■ Rol

- stocarea tuturor elementelor cunoașterii (fapte, reguli, metode de rezolvare, euristici) specifice domeniului de aplicație, preluate de la experții umani sau din alte surse

□ modulul de control

■ Conținut

- regulile prin care se pot obține informații noi
- algoritmi independenți de domeniu
- creierul SBR – un algoritm de deducere bazat pe BC și specific metodei de raționare
 - un program în care s-a implementat cunoașterea de control, procedurală sau operatorie, cu ajutorul căruia se exploatează baza de cunoștințe pentru efectuarea de raționamente în vederea obținerii de soluții, recomandări sau concluzii.
- depinde de complexitate și tipul cunoștințelor cu care are de-a face

■ Rol

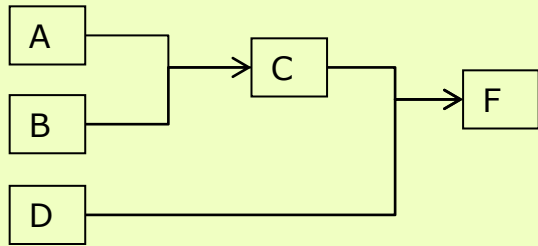
- cu ajutorul lui se exploatează baza de cunoștințe pentru efectuarea de raționamente în vederea obținerii de soluții, recomandări sau concluzii

Sisteme inteligente – SBC – sisteme de tip Bayes

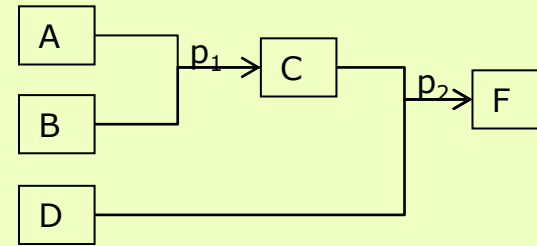
Conținut și arhitectură

□ Ideea de bază

- SBR (Sisteme expert) în care faptele și regulile sunt probabilistice
- Dacă A și B atunci C
- Dacă C și D atunci F
- Dacă A și B atunci C [cu probabilitatea p_1]
- Dacă C și D atunci F [cu probabilitatea p_2]



SBR classic



SBR de tip Bayes

Sisteme inteligente – SBC – sisteme de tip Bayes

Conținut și arhitectură

□ Regulile din BC sunt (în general) de forma:

- Dacă evenimentul (faptul) I este adevărat, atunci evenimentul (faptul) D este adevărat [cu probabilitatea p]
- Dacă evenimentul I s-a produs, atunci evenimentul D se va produce cu probabilitatea p
 - I – ipoteza (asertiune, concluzie)
 - D – dovada (premisă) care susține ipoteza

$$p(I|D) = \frac{p(D|I)p(I)}{p(D|I)p(I) + p(D|\neg I)p(\neg I)} \quad (d)$$

■ unde:

- $p(I)$ – probabilitatea apriori ca ipoteza I să fie adevărată
- $p(D|I)$ – probabilitatea ca ipoteza I fiind adevărată să implice dovada D
- $p(\neg I)$ – probabilitatea apriori ca ipoteza I să fie falsă
- $p(D|\neg I)$ – probabilitatea găsirii dovezii D chiar dacă ipoteza I este falsă

□ Cum și cine calculează aceste probabilități? → modulul de control

Sisteme inteligente – SBC – sisteme de tip Bayes

Conținut și arhitectură

- ❑ Cum calculează MC aceste probabilități într-un SBR?

$$p(I|D) = \frac{p(D|I)p(I)}{p(D|I)p(I) + p(D|\neg I)p(\neg I)} \quad (d)$$

- utilizatorul furnizează informații privind dovezile observate
 - experții determină probabilitățile necesare rezolvării problemei
 - ❑ Probabilități apriori pentru posibile ipoteze (adevărate sau false) – $p(I)$ și $p(\neg I)$
 - ❑ Probabilitățile condiționate pentru observarea dovezii D dacă ipoteza I este adevărată $p(D|I)$, respectiv falsă $p(D|\neg I)$
 - SBR calculează probabilitatea posteriori $p(I|D)$ pentru ipoteza I în condițiile dovezilor D furnizate de utilizator
-
- ❑ Actualizare de tip Bayes
 - O tehnică de actualizare a probabilității p asociate unei reguli care susține o ipoteză pe baza dovezilor (pro sau contra)
 - Inferență (raționament) de tip Bayes

Sisteme inteligente – SBC – sisteme de tip Bayes

Conținut și arhitectură

□ Actualizare de tip Bayes

- O tehnică de actualizare a probabilității p asociate unei reguli care susține o ipoteză pe baza dovezilor (pro sau contra)

- Actualizarea poate ține cont de:

- una sau mai multe (m) ipoteze (exclusive și exhaustive)
- una sau mai multe (n) dovezi (exclusive și exhaustive)

- Cazuri:

- Mai multe ipoteze și o singură dovadă

$$p(I_i | D) = \frac{p(D | I_i)p(I_i)}{\sum_{k=1}^m p(D | I_k)p(I_k)}$$

- Mai multe ipoteze și mai multe dovezi

$$p(I_i | D_1 D_2 \dots D_n) = \frac{p(D_1 D_2 \dots D_n | I_i)p(I_i)}{\sum_{k=1}^m p(D_1 D_2 \dots D_n | I_k)p(I_k)} = \frac{p(D_1 | I_i)p(D_2 | I_i) \dots p(D_n | I_i)p(I_i)}{\sum_{k=1}^m p(D_1 D_2 \dots D_n | I_k)p(I_k)}$$

Sisteme inteligente – SBC – sisteme de tip Bayes

Conținut și arhitectură

□ Exemplu numeric

■ Pp. un SBR în care:

□ utilizatorul

- furnizează 3 dovezi condiționate independente D_1 , D_2 și D_3

□ expertul

- crează 3 ipoteze mutual exclusive și exhaustive I_1 , I_2 și I_3 și stabilește probabilitățile asociate lor – $p(I_1)$, $p(I_2)$ și $p(I_3)$
- determină probabilitățile condiționate pentru observarea fiecărei dovezi pentru toate ipotezele posibile

probabilitatea	Ipotezele		
	$i = 1$	$i = 2$	$i = 3$
$p(I_i)$	0.40	0.35	0.25
$p(D_1 I_i)$	0.30	0.80	0.50
$p(D_2 I_i)$	0.90	0.00	0.70
$p(D_3 I_i)$	0.60	0.70	0.90

Sisteme inteligente – SBC – sisteme de tip Bayes

Conținut și arhitectură

□ Exemplu numeric

- Presupunem că prima dovadă observată este D_3
- SE calculează probabilitățile posteriori $p(I_i | D_3)$ pentru toate ipotezele:

probabilitatea	Ipotezele		
	i = 1	i = 2	i = 3
$p(I_i)$	0.40	0.35	0.25
$p(D_1 I_i)$	0.30	0.80	0.50
$p(D_2 I_i)$	0.90	0.00	0.70
$p(D_3 I_i)$	0.60	0.70	0.90

$$p(I_1 | D_3) = \frac{0.60 \cdot 0.40}{0.60 \cdot 0.40 + 0.70 \cdot 0.35 + 0.90 \cdot 0.25} = 0.34$$

$$p(I_2 | D_3) = \frac{0.70 \cdot 0.35}{0.60 \cdot 0.40 + 0.70 \cdot 0.35 + 0.90 \cdot 0.25} = 0.34$$

$$p(I_3 | D_3) = \frac{0.90 \cdot 0.25}{0.60 \cdot 0.40 + 0.70 \cdot 0.35 + 0.90 \cdot 0.25} = 0.32$$

- După observarea dovezii D_3
 - Încrederea în ipoteza I_2 este aceeași cu încrederea în ipoteza I_1
 - Încrederea în ipoteza I_3 crește

Sisteme inteligente – SBC – sisteme de tip Bayes

Conținut și arhitectură

□ Exemplu numeric

■ Presupunem că a doua dovadă observată este D_1

■ SE calculează probabilitățile posteriori $p(I_i | D_1 D_3)$ pentru toate ipotezele:

probabilitatea	Ipotezele		
	i = 1	i = 2	i = 3
$p(I_i)$	0.40	0.35	0.25
$p(D_1 I_i)$	0.30	0.80	0.50
$p(D_2 I_i)$	0.90	0.00	0.70
$p(D_3 I_i)$	0.60	0.70	0.90

$$p(I_1 | D_1 D_3) = \frac{0.30 \cdot 0.60 \cdot 0.40}{0.30 \cdot 0.60 \cdot 0.40 + 0.80 \cdot 0.70 \cdot 0.35 + 0.50 \cdot 0.90 \cdot 0.25} = 0.19$$

$$p(I_2 | D_1 D_3) = \frac{0.80 \cdot 0.70 \cdot 0.35}{0.30 \cdot 0.60 \cdot 0.40 + 0.80 \cdot 0.70 \cdot 0.35 + 0.50 \cdot 0.90 \cdot 0.25} = 0.52$$

$$p(I_3 | D_1 D_3) = \frac{0.50 \cdot 0.90 \cdot 0.25}{0.30 \cdot 0.60 \cdot 0.40 + 0.80 \cdot 0.70 \cdot 0.35 + 0.50 \cdot 0.90 \cdot 0.25} = 0.29$$

■ După observarea dovezii D_1

□ Încrederea în ipoteza I_1 scade

□ Încrederea în ipoteza I_2 crește (fiind cea mai probabilă de a fi adevărată)

□ Încrederea în ipoteza I_3 crește

Sisteme inteligente – SBC – sisteme de tip Bayes

Conținut și arhitectură

□ Exemplu numeric

■ Presupunem că ultima dovadă observată este D_2

■ Se calculează probabilitățile posteriori $p(I_i | D_2 D_1 D_3)$ pentru toate ipotezele:

probabilitatea	Ipotezele		
	i = 1	i = 2	i = 3
$p(I_i)$	0.40	0.35	0.25
$p(D_1 I_i)$	0.30	0.80	0.50
$p(D_2 I_i)$	0.90	0.00	0.70
$p(D_3 I_i)$	0.60	0.70	0.90

$$p(I_1 | D_2 D_1 D_3) = \frac{0.90 \cdot 0.30 \cdot 0.60 \cdot 0.40}{0.90 \cdot 0.30 \cdot 0.60 \cdot 0.40 + 0.00 \cdot 0.80 \cdot 0.70 \cdot 0.35 + 0.70 \cdot 0.50 \cdot 0.90 \cdot 0.25} = 0.45$$

$$p(I_2 | D_2 D_1 D_3) = \frac{0.00 \cdot 0.80 \cdot 0.70 \cdot 0.35}{0.90 \cdot 0.30 \cdot 0.60 \cdot 0.40 + 0.00 \cdot 0.80 \cdot 0.70 \cdot 0.35 + 0.70 \cdot 0.50 \cdot 0.90 \cdot 0.25} = 0.00$$

$$p(I_3 | D_2 D_1 D_3) = \frac{0.70 \cdot 0.50 \cdot 0.90 \cdot 0.25}{0.90 \cdot 0.30 \cdot 0.60 \cdot 0.40 + 0.00 \cdot 0.80 \cdot 0.70 \cdot 0.35 + 0.70 \cdot 0.50 \cdot 0.90 \cdot 0.25} = 0.55$$

■ După observarea dovezii D_2

- Încrederea în ipoteza I_1 crește
- Încrederea în ipoteza I_2 e nulă (ipoteza e falsă)
- Încrederea în ipoteza I_3 crește

Sisteme inteligente – SBC – sisteme de tip Bayes

Conținut și arhitectură

□ Exemplu practic

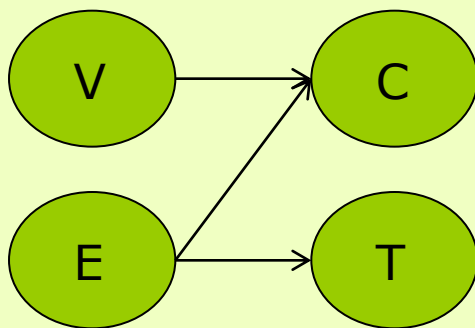
- Presupunem cazul unei mașini care nu pornește când este accelerată, dar scoate fum
 - Dacă scoate fum, atunci accelerația este defectă [cu probabilitatea 0.7]
 - $P(I_1|D_1) = 0.7$
- Pe baza unor observări statistice, experții au constatat:
 - următoarea regulă:
 - Dacă accelerația este defectă, atunci mașina scoate fum [cu probabilitatea 0.85]
 - probabilitatea ca mașina să pornească din cauză că accelerația este defectă = 0.05 (probabilitate *apriori*)
 - deci avem
 - 2 ipoteze:
 - I_1 : accelerația este defectă
 - I_2 : accelerația nu este defectă
 - o dovadă
 - D_1 : mașina scoate fum
 - probabilitatea că accelerația este defectă dacă mașina scoate fum
 - $P(I_1|D_1) = p(D_1|I_1) * p(I_1) / (p(D_1|I_1) * p(I_1) + p(D_1|I_2) * p(I_2))$
 - $P(I_1|D_1) = 0.23 < 0.7$

	I_1	I_2
$p(I_i)$	0.05	1-0.05=0.95
$P(D_1 I_i)$	0.85	1-0.85=0.15

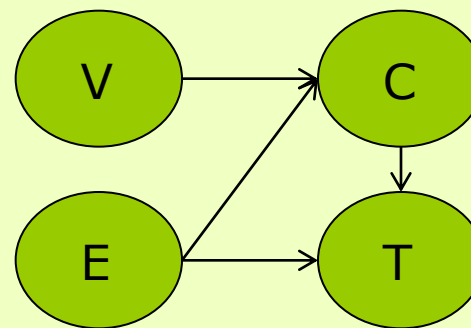
Sisteme inteligente – SBC – sisteme de tip Bayes

Tipologie

- ❑ Sisteme simple de tip Bayes
 - Consecințele unei ipoteze nu sunt corelate
 - ❑ Rețele de tip Bayes
 - Consecințele unei ipoteze pot fi corelate
- ❑ De exemplu, reținem informații despre vârsta (V), educația (E), câștigurile (C) și preferința pentru teatru (T) ale unor persoane



Sistem Bayes simplu (naiv)



Rețea Bayes simplu

Sisteme inteligente – SBC – sisteme de tip Bayes

□ Tool-uri

- MSBNx – [view](#)
- JavaBayes – [view](#)
- BNJ – [view](#)

Sisteme inteligente – SBC – sisteme de tip Bayes

- Avantaje ale inferenței de tip Bayes
 - Tehnică bazată pe teoreme statistice
 - Probabilitatea dovezilor (simptomelor) în ipotezele (cauzele) date sunt posibil de furnizat
 - Probabilitatea unei ipoteze se poate modifica datorită uneia sau mai multor dovezi

- Dezavantaje ale inferenței de tip Bayes
 - Trebuie cunoscute (sau ghicite) probabilitățile apriori ale unor ipoteze

Sisteme inteligente – SBC

- Tehnici de raționare în medii nesigure
 - Teoria Bayesiană – metodă probabilistică
 - **Teoria certitudinii**
 - Teoria posibilității (logica fuzzy)
- } Metode euristice

Sisteme inteligente – SBC – factori de certitudine

- Conținut și arhitectură
- Tipologie
- Tool-uri
- Avantaje și dezavantaje

Sisteme inteligente – SBC – factori de certitudine

Conținut și arhitectură

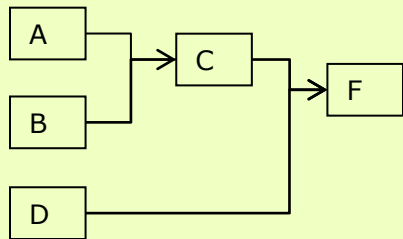
■ Ideea de bază

- SBR (sisteme expert) în care faptele și regulile au asociate câte un factor de certitudine (FC)/coeficient de încredere
- Un fel de sisteme de tip Bayes în care probabilitățile sunt înlocuite cu factori de certitudine

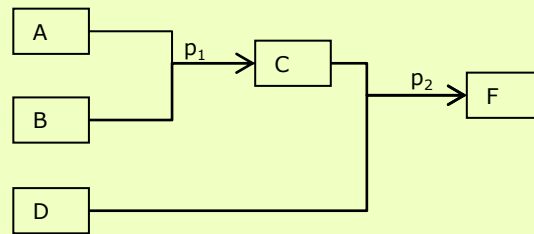
- Dacă A și B atunci C
- Dacă C și D atunci F

- Dacă A și B atunci C [cu prob p_1]
- Dacă C și D atunci F [cu prob p_2]

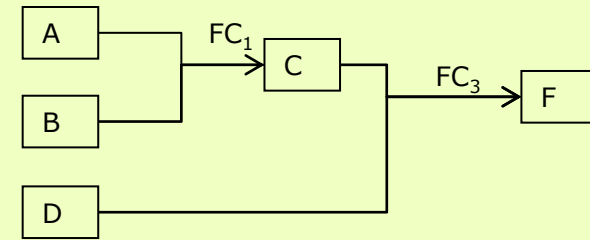
- Dacă A și B atunci C [FC_1]
- Dacă C și D [FC_2] atunci F [FC_3]



SBR classic



SBR de tip Bayes



SBR cu FC

Sisteme inteligente – SBC – factori de certitudine

Conținut și arhitectură

- FC măsoară încrederea acordată unor fapte sau reguli
- Utilizarea FC → alternativă la actualizarea de tip Bayes

- FC pot fi aplicați
 - faptelor
 - regulilor (concluziei/concluziilor unei reguli)
 - fapte + reguli

- Într-un SBR (sistem expert) cu factori de certitudine
 - regulile sunt de forma:
 - dacă dovada atunci ipoteza [FC]
 - dacă dovada_[FC] atunci ipoteza
 - dacă dovada_[FC] atunci ipoteza [FC]
 - ipotezele susținute de probe sunt independente

Conținut și arhitectură

□ FC – mod de calcul

■ Măsura încrederii (measure of belief – MB)

- măsura creșterii încrederii în ipoteza I pe baza dovezii D

$$MB(I, D) = \begin{cases} 1, & \text{dacă } p(I) = 1 \\ \frac{p(I|D) - p(I)}{1 - p(I)} & \text{dacă } p(I) < 1 \end{cases}$$

■ Măsura neîncrederii (measure of disbelief – MD)

- măsura creșterii neîncrederii în ipoteza I pe baza dovezii D

$$MD(I, D) = \begin{cases} 1, & \text{dacă } p(I) = 0 \\ \frac{p(I) - p(I|D)}{p(I)} & \text{dacă } p(I) > 0 \end{cases}$$

■ Pentru evitarea valorilor negative ale MB și MD:

$$MB(I, D) = \begin{cases} 1, & \text{dacă } p(I) = 1 \\ \frac{\max\{p(I|D), p(I)\} - p(I)}{1 - p(I)} & \text{dacă } p(I) < 1 \end{cases} \quad MD(I, D) = \begin{cases} 1, & \text{dacă } p(I) = 0 \\ \frac{\min\{p(I|D), p(I)\} - p(I)}{0 - p(I)} & \text{dacă } p(I) > 0 \end{cases}$$

■ FC – încrederea în ipoteza I dată fiind dovada D

- Număr din $[-1, 1]$
- FC = -1 dacă se știe că ipoteza I este falsă
- FC = 0 dacă nu se știe nimic despre ipoteza I
- FC = 1 dacă se știe că ipoteza I este adevărată

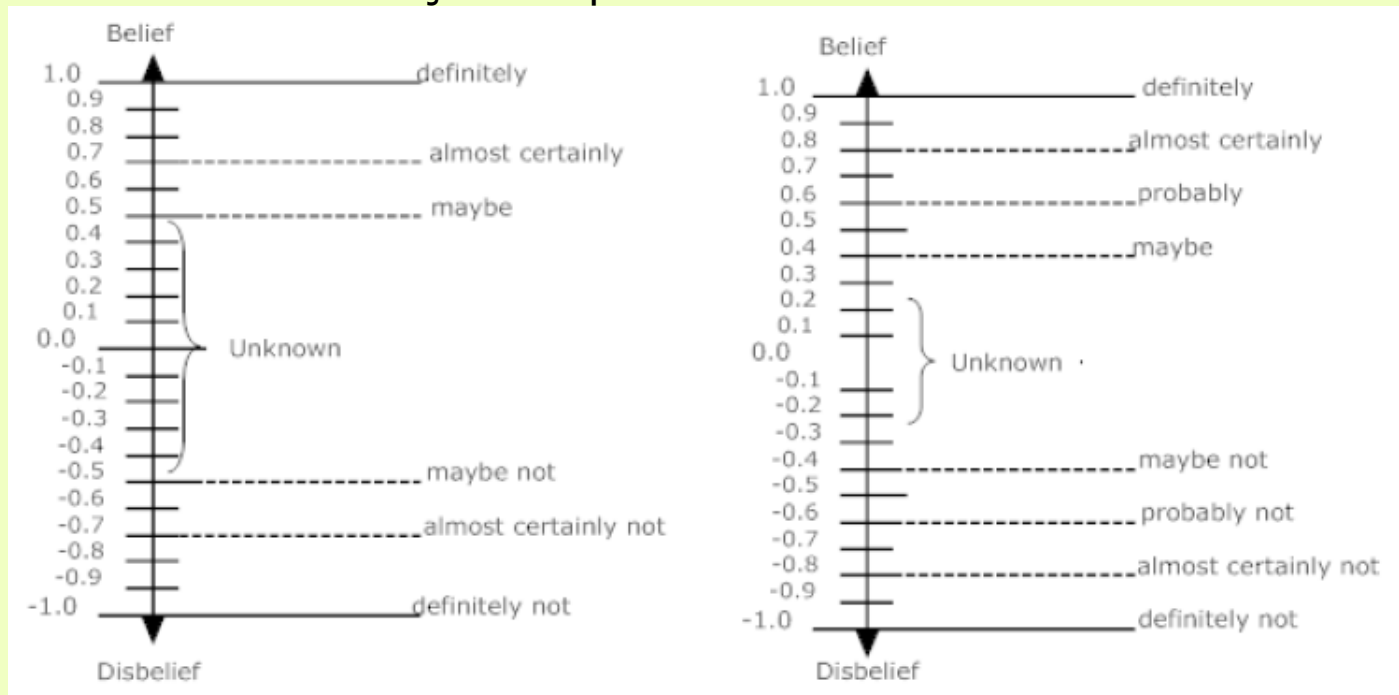
$$FC(I, D) = \frac{MB(I, D) - MD(I, D)}{1 - \min\{MB(I, D), MD(I, D)\}}$$

Sisteme inteligente – SBC – factori de certitudine

Conținut și arhitectură

FC – mod de calcul

- FC – încrederea în ipoteza I dată fiind dovada D
 - FC=-1 dacă se știe că ipoteza este falsă
 - FC=0 dacă nu se știe nimic despre ipoteză
 - FC=1 dacă se știe că ipoteza este adevărată



Conținut și arhitectură

□ FC – mod de calcul

- încrederea în ipoteza I dată fiind dovada D

□ ipoteza I poate fi:

- simplă (ex. *Dacă D atunci I*)
- compusă (ex. *Dacă D atunci I_1 și I_2 și ... I_n*)

□ dovada D poate fi

- dpdv al compoziției:
 - simplă (ex. *Dacă D atunci I*)
 - compusă (ex. *Dacă D_1 și D_2 și ... D_n atunci I*)
- dpdv al incertitudinii (încrederii în dovadă):
 - sigură (ex. *Dacă D atunci I*)
 - nesigură (ex. *Dacă $D[FC]$ atunci I*)

Conținut și arhitectură

- ❑ FC – mod de calcul pentru combinarea încrederii
 - ❑ o dovadă incertă care susține sigur o ipoteză
 - ❑ mai multe dovezi incerte care susțin sigur o singură ipoteză
 - ❑ o dovadă incertă care susține incert o ipoteză
 - ❑ mai multe dovezi incerte care susțin incert o ipoteză

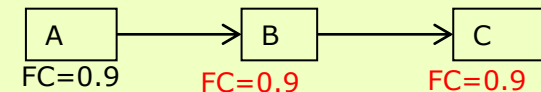
Conținut și arhitectură

- FC – mod de calcul pentru combinarea încrederii
 - O dovadă incertă care susține sigur o ipoteză

$$FC(I) = \begin{cases} FC(D), & \text{dacă } FC(D) > 0 \\ 0, & \text{altfel} \end{cases}$$

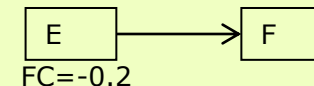
Exemplul 1

- R_1 : Dacă $A_{[FC=0.9]}$ atunci B
- R_2 : Dacă B atunci C
- $FC(B) = FC(A) = 0.9$
- $FC(C) = FC(B) = 0.9$



Exemplul 2

- R_1 : Dacă $E_{[FC=-0.2]}$ atunci F
- $FC(E \text{ este adevărat}) = -0.2 \rightarrow$ dovadă negativă \rightarrow nu putem spune nimic despre faptul că E este adevărat \rightarrow nu se poate spune nimic despre F



Sisteme inteligente – SBC – factori de certitudine

Conținut și arhitectură

FC – mod de calcul pentru combinarea încrederii

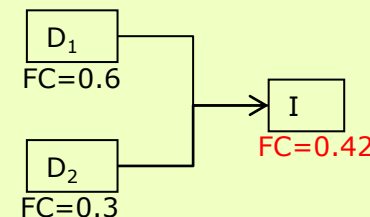
□ Mai multe dovezi incerte care susțin sigur o singură ipoteză

- Dovezi (probe) adunate incremental
- Mai multe reguli care, pe baza unor dovezi diferite, furnizează aceeași concluzi
- Aceeași ipoteză (valoare de atribut) I este obținută pe două căi de deducție distincte, cu două perechi diferite de valori pentru FC, $FC[I, D_1]$ și $FC[I, D_2]$
- Cele doua cai de deductie distincte, corespunzatoare dovezilor (probelor) D_1 și D_2 pot fi:
 - ramuri diferite ale arborelui de cautare generat prin aplicarea regulilor
 - dovezi (probe) indicate explicit sistemului

$$FC(I, D_1 \wedge D_2) = \begin{cases} CF(D_1) + CF(D_2)(1 - CF(D_1)), & \text{dacă } CF(D_1), CF(D_2) > 0 \\ CF(D_1) + CF(D_2)(1 + CF(D_1)), & \text{dacă } CF(D_1), CF(D_2) < 0 \\ \frac{CF(D_1) + CF(D_2)}{1 - \min\{|CF(D_1)|, |CF(D_2)|\}}, & \text{dacă } sign(CF(D_1)) \neq sign(CF(D_2)) \end{cases}$$

□ Exemplu

- R_1 : Dacă D_1 [$FC=0.6$] atunci I
- R_2 : Dacă D_2 [$FC=-0.3$] atunci I
- $FC(I, D_1 \wedge D_2) = (0.6 + (-0.3)) / (1 - 0.3) = 0.42$



Sisteme inteligente – SBC – factori de certitudine

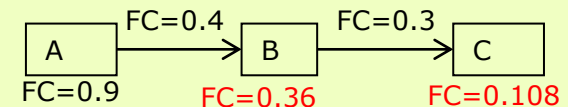
Conținut și arhitectură

- FC – mod de calcul pentru combinarea încrederii
 - O dovadă incertă care susține incert o ipoteză

$$FC(I) = \begin{cases} FC(D) * FC(\text{regulă}), & \text{dacă } FC(D) > 0 \\ 0, & \text{altfel} \end{cases}$$

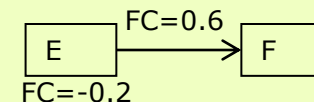
Exemplul 1

- R_1 : Dacă $A_{[FC=0.9]}$ atunci $B [FC=0.4]$
- R_2 : Dacă B atunci $C [FC=0.3]$
- $FC(B) = FC(A) * FC(R_1) = 0.9 * 0.4 = 0.36$
- $FC(C) = FC(B) * FC(R_2) = 0.36 * 0.3 = 0.108$



Exemplul 2

- R_1 : Dacă $E_{[FC=-0.2]}$ atunci $F [FC=0.6]$
- $FC(E \text{ este adevărat}) = -0.2 \rightarrow$ dovadă negativă \rightarrow nu putem spune nimic despre faptul că E este adevărat \rightarrow nu se poate spune nimic despre F



Sisteme inteligente – SBC – factori de certitudine

Conținut și arhitectură

FC – mod de calcul pentru combinarea încrederii

- Mai multe dovezi incerte care susțin incert o ipoteză
 - Dovezile sunt legate prin ȘI logic

$$CF(I) = \begin{cases} \min\{CF(D_1), CF(D_2), \dots, CF(D_n)\} * CF(\text{regulă}), & \text{dacă } CF(D_i) > 0, i = 1, 2, \dots, n \\ 0, & \text{altfel} \end{cases}$$

- Una sau mai multe dintre dovezile incerte care susțin incert o ipoteză
 - Dovezile sunt legate prin SAU logic

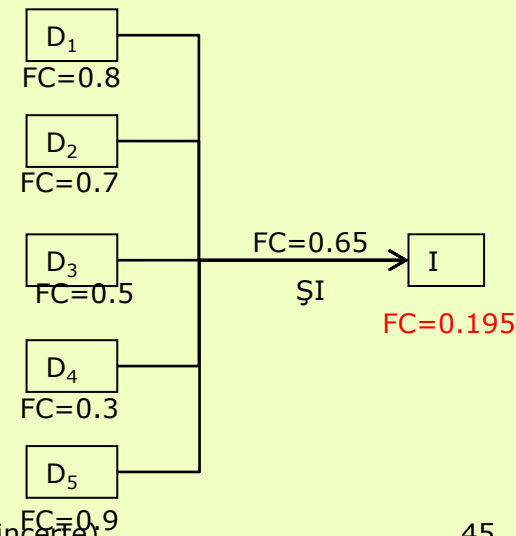
$$CF(I) = \begin{cases} \max\{CF(D_1), CF(D_2), \dots, CF(D_n)\} * CF(\text{regulă}), & \text{dacă } CF(D_i) > 0, i = 1, 2, \dots, n \\ 0, & \text{altfel} \end{cases}$$

Exemplul 1

- R_1 : Dacă D_1 _[FC = 0.8] și D_2 _[FC = 0.7] și D_3 _[FC = 0.5] și

D_4 _[FC = 0.3] și D_5 _[FC = 0.9] atunci I [FC = 0.65]

- $FC(I) = 0.3 * 0.65 = 0.195$



Sisteme inteligente – SBC – factori de certitudine

Conținut și arhitectură

FC – mod de calcul pentru combinarea încrederii

- Mai multe dovezi incerte care susțin incert o ipoteză
 - Dovezile sunt legate prin ȘI logic

$$CF(I) = \begin{cases} \min\{CF(D_1), CF(D_2), \dots, CF(D_n)\} * CF(\text{regulă}), & \text{dacă } CF(D_i) > 0, i = 1, 2, \dots, n \\ 0, & \text{altfel} \end{cases}$$

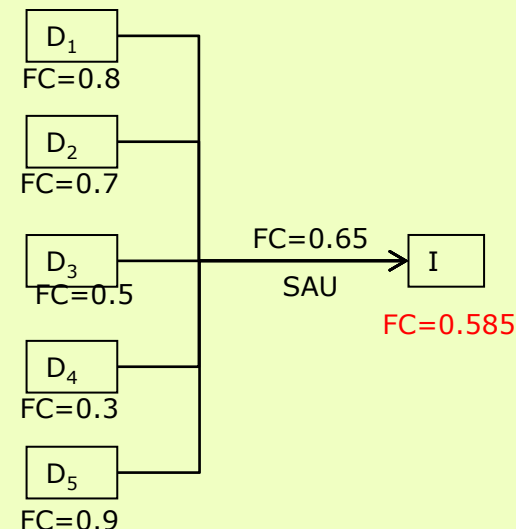
- Una sau mai multe dintre dovezile incerte susțin incert o ipoteză
 - Dovezile sunt legate prin SAU logic

$$CF(I) = \begin{cases} \max\{CF(D_1), CF(D_2), \dots, CF(D_n)\} * CF(\text{regulă}), & \text{dacă } CF(D_i) > 0, i = 1, 2, \dots, n \\ 0, & \text{altfel} \end{cases}$$

Exemplul 2

- R_1 : Dacă D_1 [FC = 0.8] sau D_2 [FC = 0.7] sau D_3 [FC = 0.5] sau D_4 [FC = 0.3] sau D_5 [FC = 0.9] atunci I [FC = 0.65]

- $FC(I) = 0.9 * 0.65 = 0.585$



Sisteme inteligente – SBC – factori de certitudine

Exemplu

□ Sistem expert pentru diagnosticarea unei răceli

■ Fapte în baza de date:

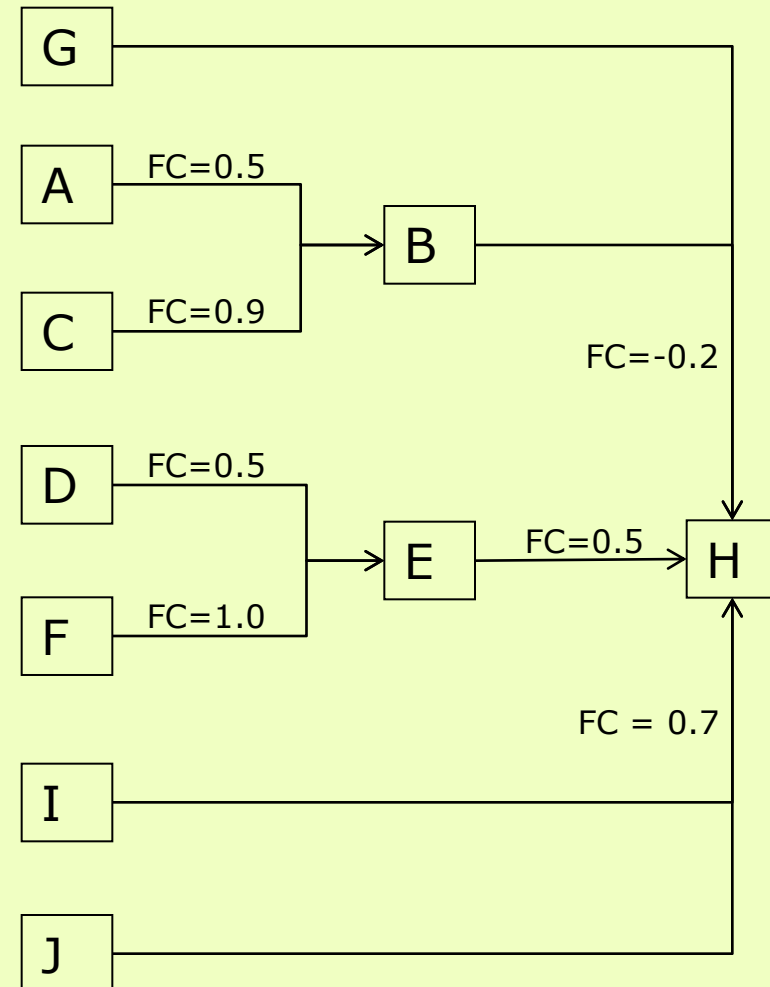
- Febra pacientului 37.4
- Pacientul tușește de mai puțin de 24 ore
- Pacientul nu are expectorații
- Pacientul are o durere de cap cu $FC = 0.4$
- Pacientul are nasul înfundat cu $FC = 0.5$

■ Reguli:

- R_1 : Dacă A : febra < 37.5 atunci
 B : simptomele de răceală sunt prezente [$FC=0.5$]
- R_2 : Dacă C : febra > 37.5 atunci
 B : simptomele de răceală sunt prezente [$FC=0.9$]
- R_3 : Dacă D : tușește > 24 ore atunci
 E : durerea de gât e prezentă [$FC=0.5$]
- R_4 : Dacă F : tușește > 48 ore atunci
 E : durerea de gât e prezentă [$FC=1.0$]
- R_5 : Dacă B : are simptome de răceală și
 G : nu expectorează atunci H : a răcit [$FC=-0.2$]
- R_6 : Dacă E : îl doare gâtul atunci
 H : a răcit [$FC=0.5$]
- R_7 : Dacă I : îl doare capul și
 J : are nasul înfundat atunci H : a răcit [$FC=0.7$]

■ Concluzia:

- Pacientul este sau nu răcit?



Sisteme inteligente – SBC – factori de certitudine

Exemplu

□ Sistem expert pentru diagnosticarea unei răceli

■ Fapte în baza de date:

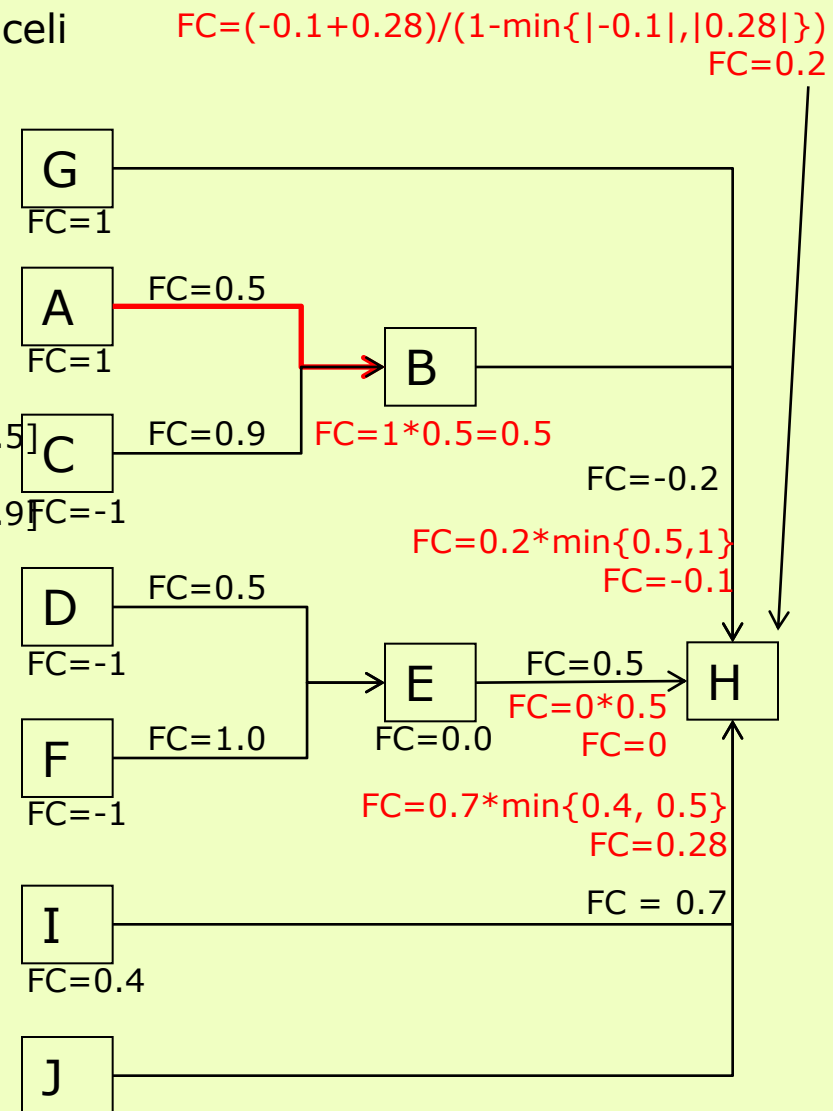
- Febra pacientului 37.4
- Pacientul tușește de mai puțin de 24 ore
- Pacientul nu are expectorații
- Pacientul are o durere de cap cu FC = 0.4
- Pacientul are nasul înfundat cu FC = 0.5

■ Reguli:

- R_1 : Dacă A: febra < 37.5 atunci
B: simptomele de răceală sunt prezente [FC=0.5]
- R_2 : Dacă C: febra > 37.5 atunci
B: simptomele de răceală sunt prezente [FC=0.9]
- R_3 : Dacă D: tușește > 24 ore atunci
E: durerea de gât e prezentă [FC=0.5]
- R_4 : Dacă F: tușește > 48 ore atunci
E: durerea de gât e prezentă [FC=1.0]
- R_5 : Dacă B: are simptome de răceală și
G: nu expectorează atunci H: a răcit [FC=-0.2]
- R_6 : Dacă E: îl doare gâtul atunci
H: a răcit [FC=0.5]
- R_7 : Dacă I: îl doare capul și
J: are nasul înfundat atunci H: a răcit [FC=0.7]

■ Concluzia:

- Pacientul este sau nu răcit?



Sisteme inteligente – SBC – factori de certitudine

□ Avantaje

- Nu este necesar calculul apriori a probabilităților

□ Limite

- ipotezele sustinute de probe sunt independente.

■ exemplu:

□ Fie următoarele fapte:

- A: Aspersorul a funcționat noaptea trecută
- U: Iarba este udă dimineață
- P: Noaptea trecută a plouat.

□ și următoarele două reguli care leagă între ele aceste fapte:

- R_1 : dacă aspersorul a funcționat noaptea trecută atunci există o încredere puternică (0.9) că iarba este udă dimineața
- R_2 : dacă iarba este udă dimineața atunci există o încredere puternică (0.8) că noaptea trecută a plouat

□ Deci:

- $FC[U,A] = 0.9$ - deci proba aspersor sustine iarba uda cu 0.9
- $FC[P,U] = 0.8$ - deci iarba uda sustine ploaie cu 0.8
- $FC[P,A] = 0.8 * 0.9 = 0.72$ - deci aspersorul sustine ploaia cu 0.72

Sisteme inteligente – SBC – factori de certitudine

□ SBR de tip Bayes vs. SBR cu FC

Bayes	FC
Teorie probabilităților este veche și fundamentată matematic	Teoria FC este nouă și fără demonstrații matematice
Necesită existența unor informații statistice	Nu necesită existența unor date statistice
Propagarea încrederii crește în timp exponențial	Informația circulă repede și eficient în SBR

□ Tehnici de raționare în medii nesigure

- Teoria Bayesiană – metodă probabilistică

- Teoria certitudinii

- **Teoria posibilității (logica fuzzy)**



Metode
euristice

Sisteme inteligente – SBC – sisteme fuzzy

- Teoria posibilității
- Conținut și arhitectură
- Tipologie
- Tool-uri
- Avantaje și limite

Sisteme inteligente – SBC – sisteme fuzzy

Teoria posibilității (logica fuzzy)

□ De ce fuzzy?

- Problemă: transpuneți în cod (C++) următoarele propoziții:
 - Georgel este înalt.
 - Afara este frig.

□ Când este important fuzzy?

- Interogări în limbaj natural
- Reprezentarea cunoștințelor în sisteme expert
- Controlul fuzzy – când se lucrează cu fenomene imprecise (perturbate de zgomot)

Sisteme inteligente – SBC – sisteme fuzzy

Amintim componența unui SBC

- Baza de cunoștințe (BC) → Modalități de reprezentare a cunoștințelor
 - Logica formală (limbaje formale)
 - Definiție
 - Știința principiilor formale de raționament
 - Componente
 - Sintaxă
 - Semantică
 - Metodă de inferență sintactică
 - Tipologie
 - În funcție de numărul valorilor de adevăr:
 - logică duală
 - **logică polivalentă**
 - În funcție de tipul elementelor de bază:
 - clasică → primitivele = propoziții (predicate)
 - probabilistică → primitivele = variabile aleatoare
 - În funcție de obiectul de lucru:
 - logica propozițională → se lucrează doar cu propoziții declarative, iar obiectele descrise sunt fixe sau unice (Ionică este student)
 - logica predicatelor de ordin I → se lucrează cu propoziții declarative, cu predicate și cuantificări, iar obiectele descrise pot fi unice sau variabile asociate unui obiect unic (Toți studenții sunt prezenți)
 - Reguli
 - Rețele semantice
- Modulul de control (MC – pentru inferență)

Sisteme inteligente – SBC – sisteme fuzzy

Teoria posibilității - Un pic de istorie

- ❑ Parminedes (400 B.C.)
- ❑ Aristotle
 - "Law of the Excluded Middle" – fiecare propoziție trebuie să fie Adevărată sau Falsă
- ❑ Plato
 - O a treia regiune între Adevărat și Fals
 - Pune bazele logicii fuzzy
- ❑ Lukasiewicz (1900)
 - Propune o alternativă sistematică la logica bivalentă a lui Aristotle – logica trivalentă: Adevărat, Fals, Posibil
- ❑ Lotfi A. Zadeh (1965)
 - Descrie matematic teoria mulțimilor fuzzy și logica fuzzy: funcția de apartenență (valorile Adevărat și Fals) operează pe intervalul $[0,1]$
 - ❑ A propus noi operații de calcul pt logica fuzzy
 - ❑ A considerat logica fuzzy o generalizare a logicii clasice
 - A publicat primul articol despre mulțimile fuzzy

Teoria posibilității

□ Logica fuzzy

- Generalizare a logicii Boolene
- Manipulează conceptul de adevăr parțial
 - Logica clasică – totul este exprimat în termeni binari
 - 0 sau 1, alb sau negru, da sau nu
 - Logica fuzzy – exprimarea graduală a unui adevăr
 - Valori între 0 și 1

□ Analogia *logică vs. algebră*

- Operatorii logici exprimați în termeni matematici (George Boole):
 - Conjuncție = minimum $\rightarrow a \wedge b = \min(a, b)$
 - Disjuncție = maximum $\rightarrow a \vee b = \max(a, b)$
 - Negație = scădere $\rightarrow \neg a = 1 - a$

Sisteme inteligente – SBC – sisteme fuzzy

Reamintim: SBR – arhitectură

□ baza de cunoștințe

■ Conține

- Informațiile specifice despre un domeniu sub forma unor
 - fapte – afirmații corecte
 - reguli - euristici speciale care generează informații (cunoștințe)

■ Rol

- stocarea tuturor elementelor cunoașterii (fapte, reguli, metode de rezolvare, euristici) specifice domeniului de aplicație, preluate de la experții umani sau din alte surse

□ modulul de control

■ Conținut

- regulile prin care se pot obține informații noi
- algoritmi independenți de domeniu
- creierul SBR – un algoritm de deducere bazat pe BC și specific metodei de raționare
 - un program în care s-a implementat cunoașterea de control, procedurală sau operatorie, cu ajutorul căruia se exploatează baza de cunoștințe pentru efectuarea de raționamente în vederea obținerii de soluții, recomandări sau concluzii.
- depinde de complexitate și tipul cunoștințelor cu care are de-a face

■ Rol

- cu ajutorul lui se exploatează baza de cunoștințe pentru efectuarea de raționamente în vederea obținerii de soluții, recomandări sau concluzii

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură

□ Ideea de bază

■ Cf. teoriei informațiilor certe

- *Popescu este tânăr*

■ Cf. teoriei informațiilor incerte

- Cf. teoriei probabilităților:

- *Există 80% șanse ca Popescu să fie tânăr*

- Cf. logicii fuzzy:

- *Gradul de apartenență al lui Popescu la grupul de oameni tineri este 0.80*

□ Necesitate

■ Abordarea fenomenelor reale implică utilizarea mulțimilor fuzzy

■ Exemplu

- *Temperatura unei camere poate fi:*

- *joasă,*
- *medie sau*
- *ridicată*

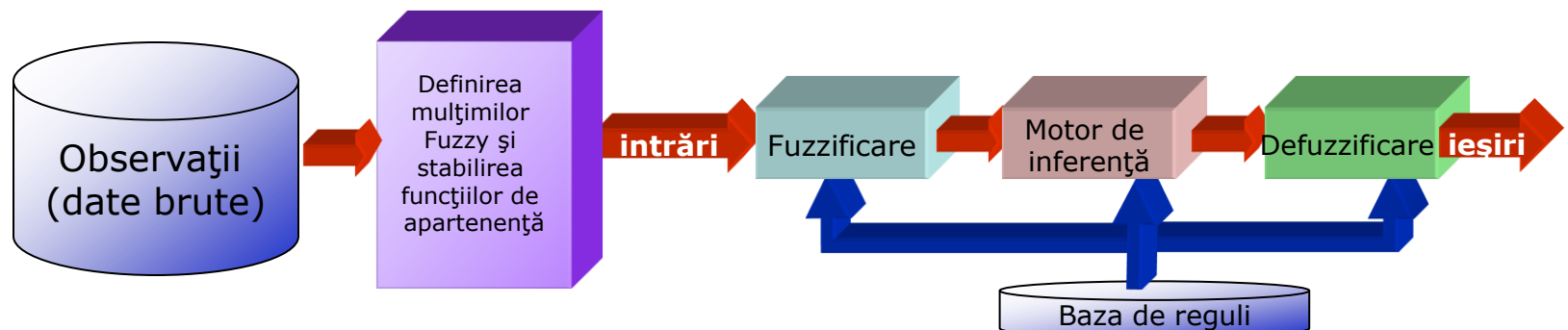
- Aceste mulțimi de temperaturi posibile se pot suprapune

- o temperatura poate aparține uneia sau mai multor mulțimi în funcție de cel care face evaluarea

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură

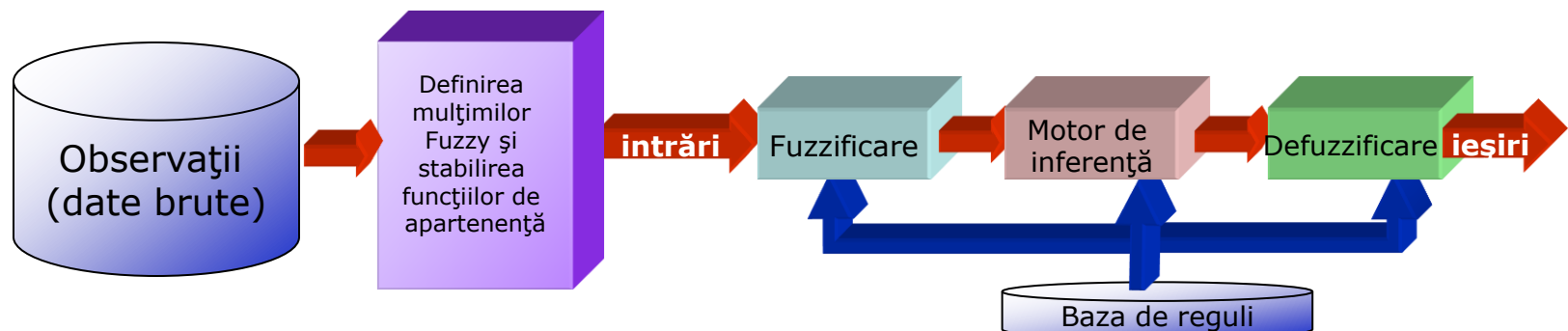
- Pași în construirea unui sistem fuzzy
 - Definirea intrărilor și ieșirilor de către expert
 - Datele de intrare și ieșire brute
 - Fuzzificarea datelor de intrare și ieșire
 - Stabilirea variabilelor fuzzy și a mulțimilor fuzzy pe baza funcțiilor de apartenență
 - Construirea unei baze de reguli de către expert
 - Matricea de decizie a bazei de cunoștințe
 - Evaluarea regulilor
 - Inferența – transformarea intrărilor fuzzy în ieșiri fuzzy prin aplicarea regulilor din baza de cunoștințe
 - Agregarea rezultatelor
 - Defuzificarea
 - Interpretarea rezultatelor



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură

- Pași în construirea unui sistem fuzzy
 - Definirea intrărilor și ieșirilor de către expert
 - Datele de intrare și ieșire brute
 - Fuzzificarea datelor de intrare și ieșire
 - Stabilirea variabilelor fuzzy și a mulțimilor fuzzy pe baza funcțiilor de apartenență
 - **Construirea unei baze de reguli de către expert**
 - **Matricea de decizie a bazei de cunoștințe**
 - Evaluarea regulilor
 - Inferența – transformarea intrărilor fuzzy în ieșiri fuzzy prin aplicarea regulilor din baza de cunoștințe
 - Agregarea rezultatelor
 - Defuzificarea
 - Interpretarea rezultatelor



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

- Elemente de teoria posibilității (logica fuzzy)
 - Fapte (mulțimi) fuzzy
 - Definiție
 - Reprezentare
 - Operații – complement, containment, intersecție, reuniune, egalitate, produs algebric, sumă algebrică
 - Proprietăți – asociativitate, distributivitate, comutativitate, tranzitivitate, idempotenți, identitate, involuție, legile lui De Morgan
 - Hedges
 - Variabile fuzzy
 - Definiție
 - Proprietăți
- Stabilirea variabilelor fuzzy și a mulțimilor fuzzy pe baza funcțiilor de apartenență

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **definire**

■ Definierea oricărei mulțimi – 2 moduri:

□ Prin enumerarea elementelor

▪ Ex. *Mulțimea elevilor* = {Ana, Maria, Ioana}

□ Prin specificare unei proprietăți a elementelor mulțimii

▪ Ex. *Mulțimea numerelor pare* = { $x \mid x = 2n$, unde n - număr natural}

■ Funcția caracteristică μ a unei mulțimi

□ Fie X o mulțime universală și x un element al mulțimii ($x \in X$)

□ Logica clasică

▪ Fie R o submulțime a lui X : $R \subset X$, R - mulțime regulată

▪ Orice element x aparține sau nu mulțimii R

▪ $\mu_R : X \rightarrow \{0, 1\}$, unde $\mu_R(x) = \begin{cases} 1, & x \in R \\ 0, & x \notin R \end{cases}$

□ Logica fuzzy

▪ Fie F o submulțime a lui X (univers de discurs): $F \subset X$, F - mulțime fuzzy

▪ orice element x aparține mulțimii F într-un anumit grad $\mu_F(x)$

▪ $\mu_F : X \rightarrow [0, 1]$, $\mu_F(x) = g$, unde $g \in [0, 1]$ - gradul de apartenență al lui x la F

▪ $g = 0 \rightarrow$ ne-apartență

▪ $g = 1 \rightarrow$ apartenență

▪ O mulțime fuzzy = o pereche (F, μ_F) , unde $\mu_F(x) = \begin{cases} 1, & \text{dacă } x \text{ este total în } F \\ 0, & \text{dacă } x \text{ nu este în } F \\ \in (0,1) & \text{dacă } x \text{ este parte din } F \text{ (} x \text{ este număr fuzzy)} \end{cases}$

Sisteme inteligente – SBC – sisteme fuzzy

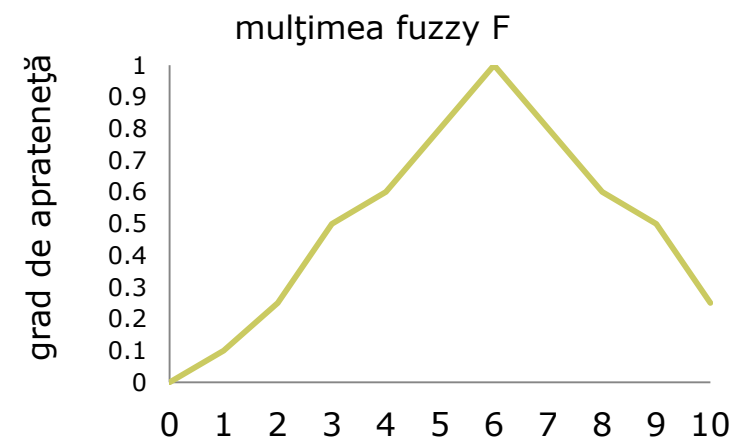
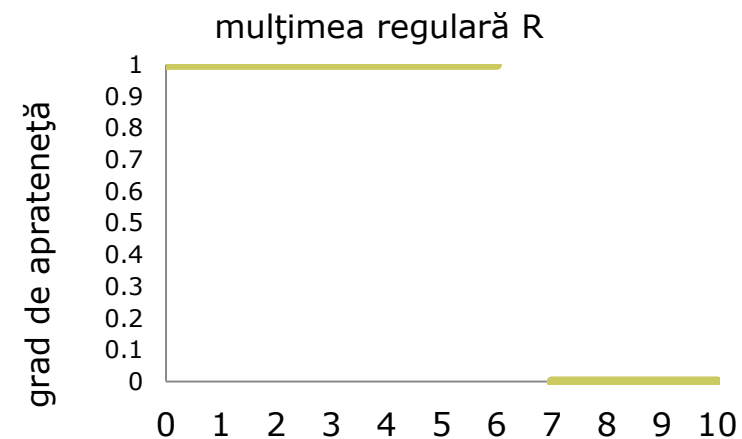
Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **definire**

■ Exemplul 1

- X – mulțimea numerelor naturale mai mici decât 11
- R – mulțimea numerelor mai mici decât 7
- F – mulțimea numerelor apropiate de 6

x	$\mu_R(x)$	$\mu_F(x)$
0	1	0
1	1	0.1
2	1	0.25
3	1	0.5
4	1	0.6
5	1	0.8
6	1	1
7	0	0.8
8	0	0.6
9	0	0.5
10	0	0.25



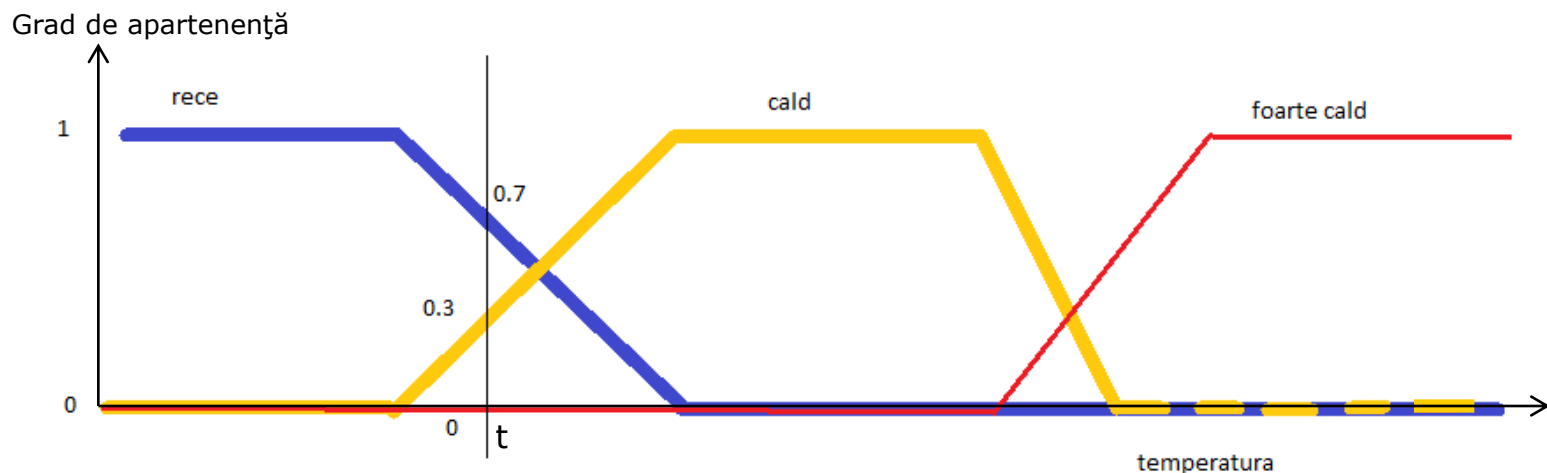
Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **definire**

■ Exemplul 2

- O anumită temperatură t poate avea 3 valori de adevăr:
 - Roșu (0): nu e fierbinte
 - Portocalui (0.3): puțin cald
 - Albastru (0.7): aproape rece



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **reprezentare**

■ Mulțimile regulate

□ Delimitări clare (exacte) → diagrame Venn

■ Mulțimile fuzzy

□ Delimitări graduale → reprezentări bazate pe funcția de apartenență

■ Singulară

■ $\mu(x) = s$, unde s este un scalar

■ Triunghiulară

$$\mu(x) = \max\left\{0, \min\left\{\frac{x-a}{b-a}, 1, \frac{c-x}{c-b}\right\}\right\}$$

■ Trapezoidală

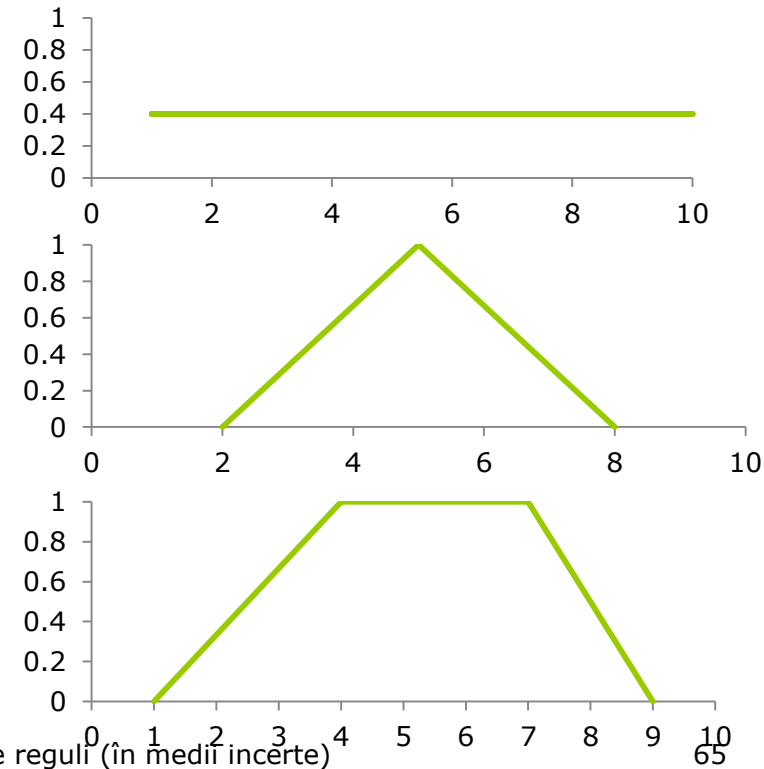
$$\mu(x) = S(x) = \max\left\{0, \min\left\{\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right\}\right\}$$

■ Funcția Z

■ $\mu(x) = 1 - S(x)$

■ Funcția Π

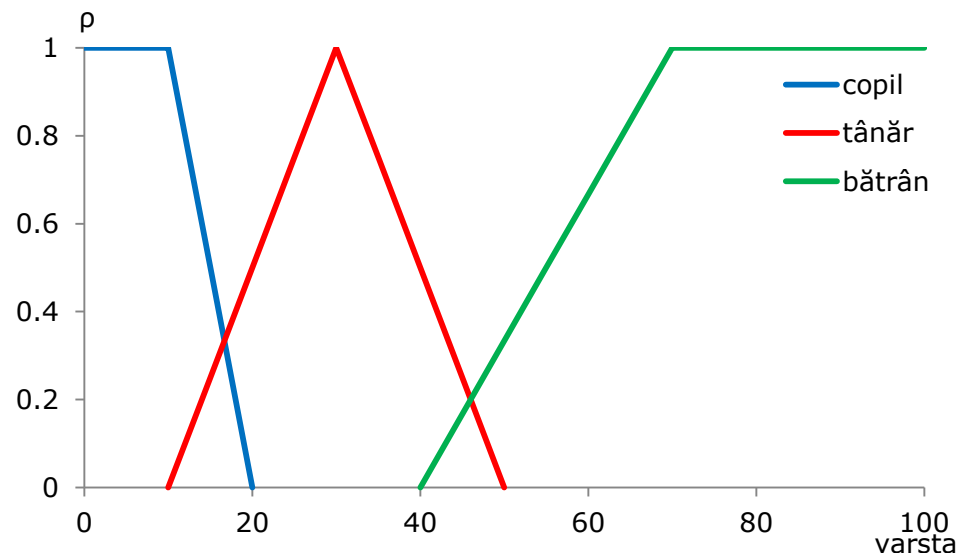
$$\mu(x) = \Pi(x) = \begin{cases} S(x), & \text{dacă } x \leq c \\ Z(x), & \text{dacă } x > c \end{cases}$$



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

- Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **reprezentare**
 - Exemplu
 - *Vârsta unei persoane*



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

- Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **Operații**
 - complement
 - incluziune
 - intersecție
 - reuniune
 - egalitate
 - produs algebric
 - sumă algebrică

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **Operații**

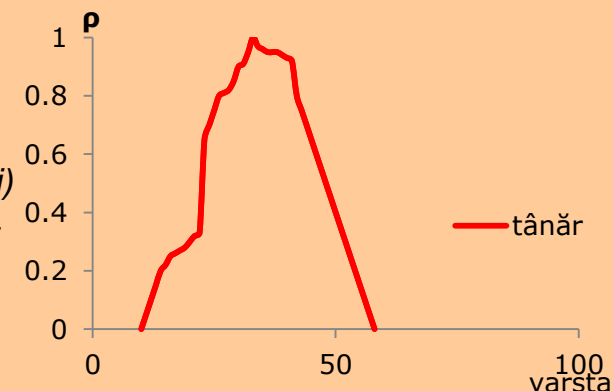
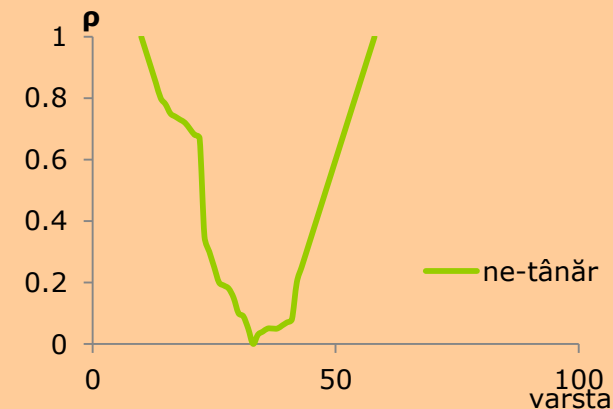
■ Complement

- X - un univers
- A – o mulțimea fuzzy (cu universul X)
- B – o mulțimea fuzzy (cu universul X)

- B este complementul lui A ($B = \neg A$) dacă:
 - $\mu_B(x) = \mu_{\neg A}(x) = 1 - \mu_A(x)$ pentru orice $x \in X$

□ Exemplu:

- *persoane bătrâne (pe baza vârstei)*
 - $A = \{(30, 0), (40, 0.2), (50, 0.4), (60, 0.6), (70, 0.8), (80, 1)\}$
- *persoane tinere (care nu sunt bătrâne) (pe baza vârstei)*
 - $\neg A = \{(30, 1), (40, 0.8), (50, 0.6), (60, 0.4), (70, 0.2), (80, 0)\}$



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **Operații**

■ Incluziune

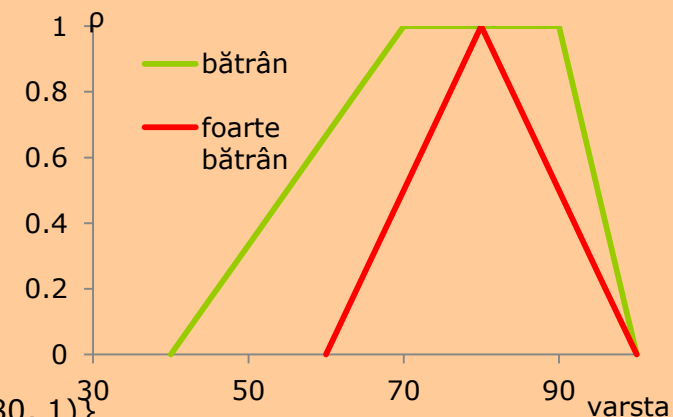
- X - un univers
- A - o mulțimea fuzzy (cu universul X)
- B - o mulțimea fuzzy (cu universul X)

□ B este submulțime a lui A ($B \subset A$) dacă:

- $\mu_B(x) \leq \mu_A(x)$ pentru orice $x \in X$

□ Exemplu

- *persoane bătrâne (pe baza vârstei)*
 - $A = \{(60, 0.6), (65, 0.7), (70, 0.8), (75, 0.9), (80, 1)\}$
- *persoane foarte bătrâne (pe baza vârstei)*
 - $B = \{(60, 0.6), (65, 0.67), (70, 0.8), (75, 0.8), (80, 0.95)\}$



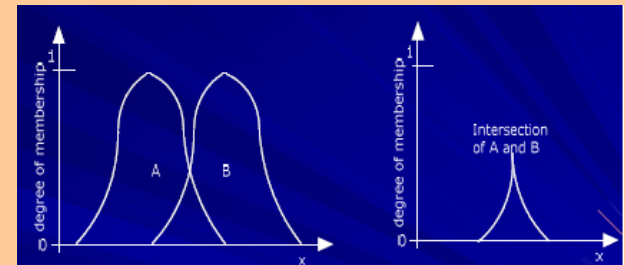
Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **Operații**

■ Intersecție

- X - un univers
- A – o mulțimea fuzzy (cu universul X)
- B – o mulțimea fuzzy (cu universul X)
- C – o mulțimea fuzzy (cu universul X)
- C este intersecția lui A cu B ($C=A \cap B$) dacă:
 - $\mu_C(x) = \mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \cap \mu_B(x)$ pentru orice $x \in X$



□ Exemplu

- *persoane bătrâne (pe baza vârstei)*
 - $A = \{(30, 0) (40, 0.1) (50, 0.2) (60, 0.6), (65, 0.7) (70, 0.8), (75, 0.9), (80, 1)\}$
- *persoane de vârstă medie*
 - $B = \{(30, 0.1) (40, 0.2) (50, 0.6) (60, 0.5), (65, 0.2) (70, 0.1), (75, 0), (80, 0)\}$
- *persoane bătrâne și de vârstă medie*
 - $C = \{(30, 0) (40, 0.1) (50, 0.2) (60, 0.5), (65, 0.2) (70, 0.1), (75, 0), (80, 0)\}$

Sisteme inteligente – SBC – sisteme fuzzy

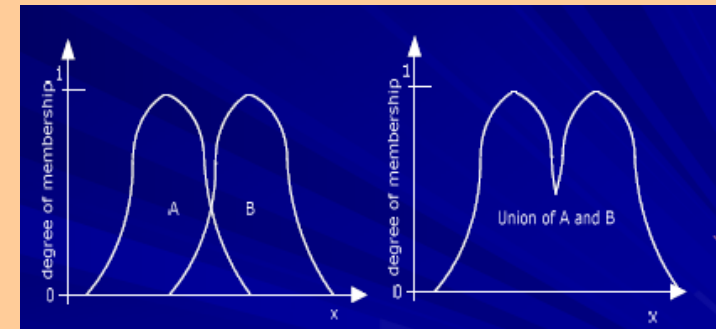
Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **Operații**

■ Reuniune

- X - un univers
- A – o mulțimea fuzzy (cu universul X)
- B – o mulțimea fuzzy (cu universul X)
- C – o mulțimea fuzzy (cu universul X)
- C este reuniunea lui A cu B ($C=A \cup B$) dacă:

- $\mu_C(x) = \mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \cup \mu_B(x)$ pentru orice $x \in X$



□ Exemplu

- *persoane bătrâne (pe baza vârstei)*
 - $A = \{(30, 0) (40, 0.1) (50, 0.2) (60, 0.6), (65, 0.7) (70, 0.8), (75, 0.9), (80, 1)\}$
- *persoane de vârstă medie*
 - $B = \{(30, 0.1) (40, 0.2) (50, 0.6) (60, 0.5), (65, 0.2) (70, 0.1), (75, 0), (80, 0)\}$
- *persoane bătrâne sau de vârstă medie*
 - $C = \{(30, 0.1) (40, 0.2) (50, 0.6) (60, 0.6), (65, 0.7) (70, 0.8), (75, 0.9), (80, 1)\}$

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **Operații**

■ Egalitate, produs și sumă algebrică

- X - un univers
- A - o mulțimea fuzzy (cu universul X)
- B - o mulțimea fuzzy (cu universul X)
- C - o mulțimea fuzzy (cu universul X)

□ B este egal cu A ($B=A$) dacă:

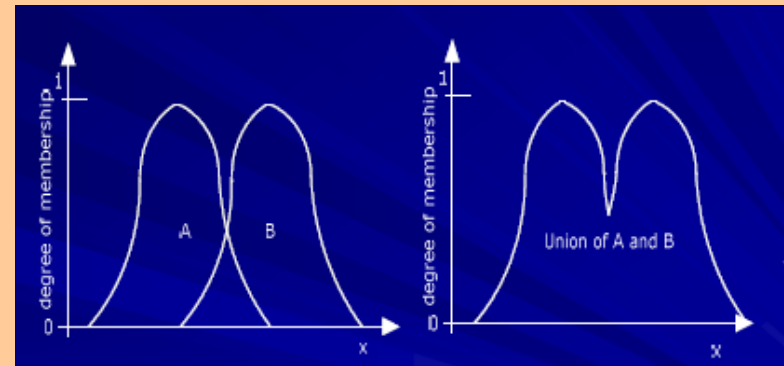
- $\mu_B(x)=\mu_A(x)$ pentru orice $x\in X$

□ C este produsul dintre A și B ($C=A*B$) dacă:

- $\mu_C(x)=\mu_{A*B}(x)=\mu_A(x)*\mu_B(x)$ pentru orice $x\in X$

□ C este suma lui A cu B ($C=A+B$) dacă:

- $\mu_C(x)=\mu_{A+B}(x)=\mu_A(x)+\mu_B(x)$ pentru orice $x\in X$



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

- Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **proprietăți**
 - Asociativitate
 - Comutativitate
 - Distributivitate
 - Tranzitivitate
 - Idempotență
 - Identitate
 - Involuție

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

- Elemente de teoria posibilității (logica fuzzy) → Fapte (mulțimi) fuzzy → **Hedges (nuanțatori)**
 - Ideea de bază
 - Modificatori, adjective sau adverbe care schimbă valorile de adevăr ale unor afirmații
 - Ex. *Foarte, mai puțin, mai mult, aproape, etc.*
 - Modifică forma mulțimilor fuzzy
 - Pot acționa la nivel de
 - Numere fuzzy
 - Valori de adevăr
 - Funcții de apartenență
 - Euristici
 - Utilitate
 - Apropierea de limbajul natural → subiectivism
 - Evaluarea variabilelor lingvistice

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) →

Fapte (mulțimi) fuzzy → **Hedges (nuanțatori)**

■ Tipologie

□ Hedge-uri care reduc valoarea de adevăr (producând o concentrație)

- *foarte* $\mu_{A_foarte}(x) = (\mu_A(x))^2$

- *extrem* $\mu_{A_extrem}(x) = (\mu_A(x))^3$

- *foarte foarte* $\mu_{A_foarte_foarte}(x) = (\mu_{A_foarte}(x))^2 = (\mu_A(x))^4$

□ Hedge-uri care măresc valoarea de adevăr (producând o dilatare)

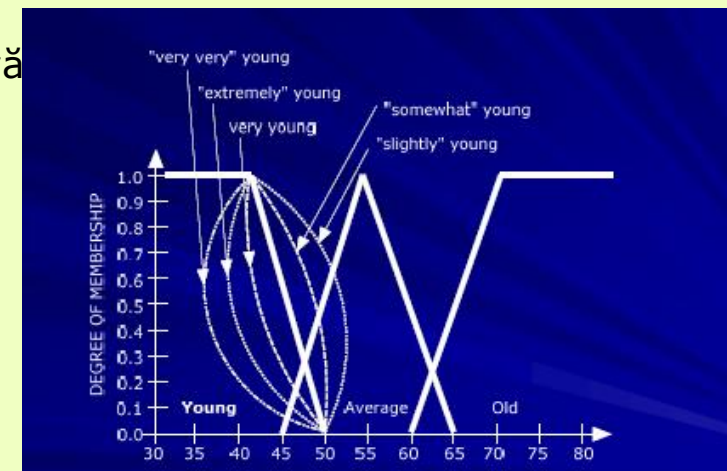
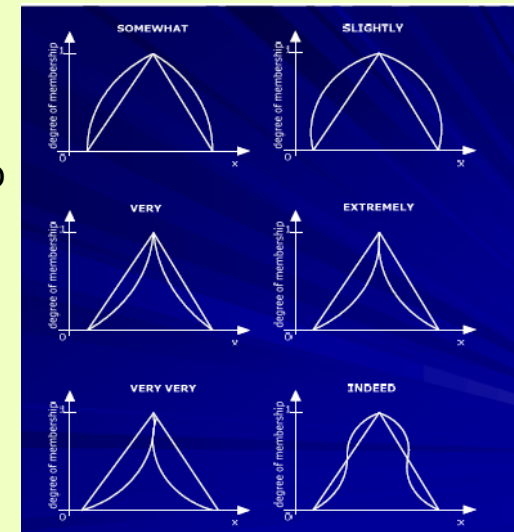
- *oarecum* $\mu_{A_oarecum}(x) = (\mu_A(x))^{1/2}$

- *ușor* $\mu_{A_ușor}(x) = (\mu_A(x))^{1/3}$

□ Hedge-uri care intensifică valoarea de adevăr

- *întradevăr*

$$\mu_{A_întradevăr}(x) = \begin{cases} 2(\mu_A(x))^2, & \text{dacă } 0 \leq \mu_A(x) \leq 0.5 \\ 1 - 2(1 - \mu_A(x))^2, & \text{dacă } 0.5 \leq \mu_A(x) \leq 1 \end{cases}$$



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

- Elemente de teoria posibilității (logica fuzzy)
 - Fapte (mulțimi) fuzzy
 - Definire
 - Reprezentare
 - Operații – complement, containment, intersecție, reuniune, egalitate, produs algebric, sumă algebrică
 - Proprietăți – asociativitate, distributivitate, comutativitate, tranzitivitate, idempotenți, identitate, involuție, legile lui De Morgan
 - Hedges
 - **Variabile fuzzy**
 - **Definiție**
 - **Proprietăți**
- Stabilirea variabilelor fuzzy și a mulțimilor fuzzy pe baza funcțiilor de apartenență

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

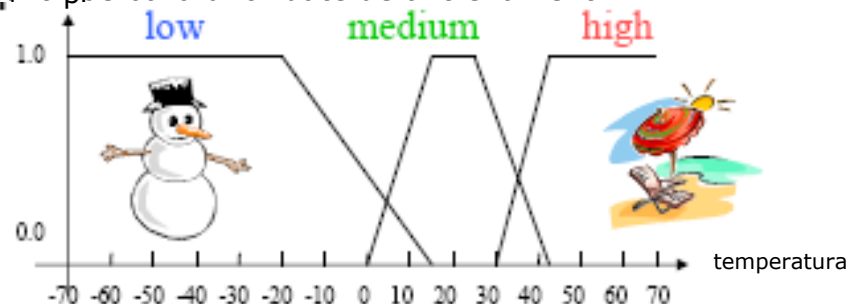
□ Elemente de teoria posibilității (logica fuzzy) → Variabile fuzzy →

Definiție

- O variabilă fuzzy V este definită de quadruplul $V = \{x, l, u, m\}$, unde:
 - X – numele variabilei simbolice
 - L – mulțimea etichetelor posibile pentru variabila x
 - U – universul variabilei
 - M – regulile semantice care definesc înțelesul fiecărei etichete din L (funcțiile de apartenență)
- Funcția de apartenență
 - Evaluare subiectivă
 - forma funcțiilor este definită de experți/specialiști
 - Evaluare ad-hoc
 - funcții simple care să poată rezolva problema
 - Evaluare bazată pe distribuții și probabilități ale informațiilor extrase din măsurători
 - Evaluare adaptată
 - prin teste
 - Evaluare automată
 - algoritmi folosiți pentru definirea funcțiilor pe baza unor date de antrenament

Exemplu

- $X =$ Temperatura
- $L = \{\text{joasă, medie, ridicată}\}$
- $U = \{x \in X \mid -70^\circ \leq x \leq +70^\circ\}$
- $M =$



Sisteme inteligente – SBC – sisteme fuzzy

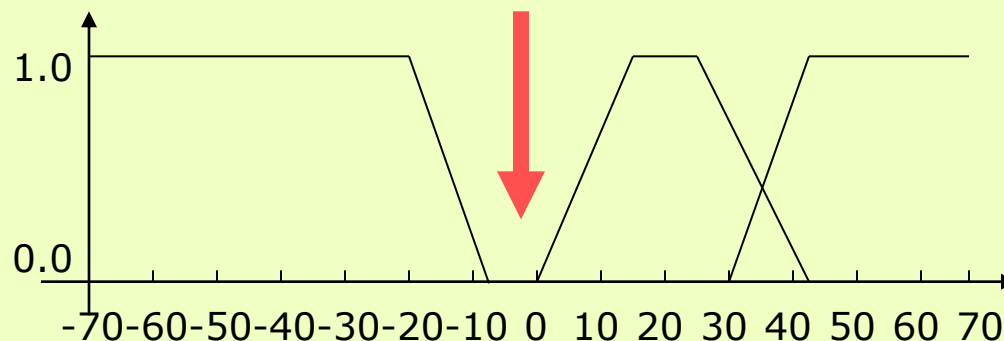
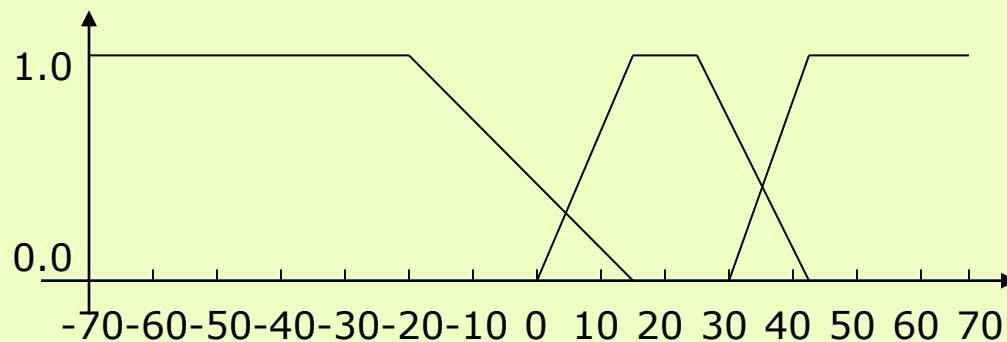
Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) → Variabile fuzzy →

Proprietăți

■ Completitudine

- O variabilă fuzzy V este completă dacă pentru orice $x \in X$ există o mulțime fuzzy A astfel încât $\mu_A(x) > 0$



Sisteme inteligente – SBC – sisteme fuzzy

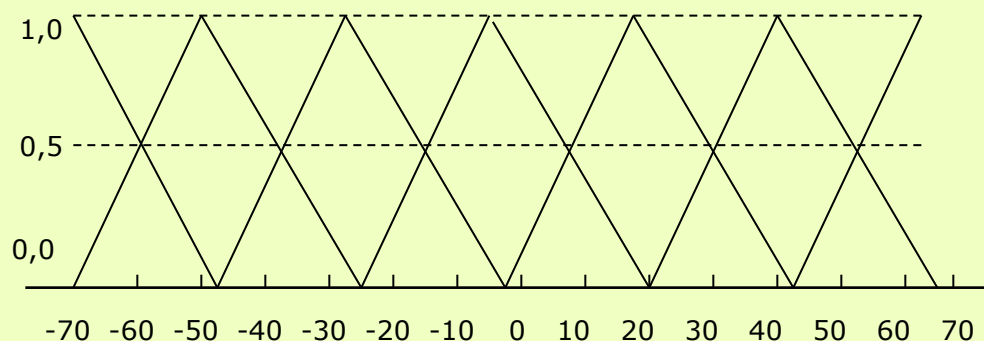
Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) → Variabile fuzzy →

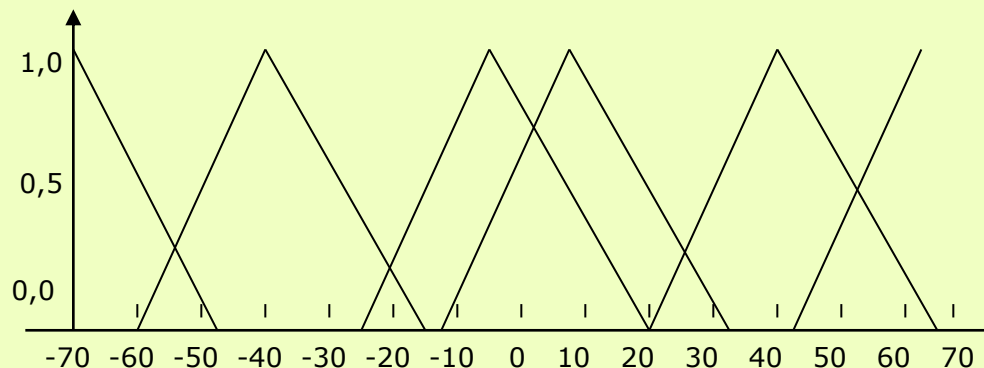
Proprietăți

■ Partiționare a unității

- O variabilă fuzzy V formează o partiție a unității dacă pentru orice valoare de intrare x avem $\sum_{i=1}^p \mu_{A_i}(x) = 1$
- Unde p este numărul mulțimilor cărora aparține x
- Nu există reguli pentru definirea suprapunerii a 2 mulțimi vecine
 - De obicei, suprapunerea trebuie să fie între 25% și 50%



Partiționare a unității



Nepartiționare a unității

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

□ Elemente de teoria posibilității (logica fuzzy) → Variabile fuzzy →

Proprietăți

■ Partiționare a unității

□ O variabilă fuzzy completă poate fi transformată într-o partiționare a unității astfel:

$$\mu_{\hat{A}_i}(x) = \frac{\mu_{A_i}(x)}{\sum_{j=1}^p \mu_{A_j}(x)} \text{ for } i = 1, K, p$$

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

- Elemente de teoria posibilității (logica fuzzy)
 - Fapte (mulțimi) fuzzy
 - Definire
 - Reprezentare
 - Operații – complement, containment, intersecție, reuniune, egalitate, produs algebric, sumă algebrică
 - Proprietăți – asociativitate, distributivitate, comutativitate, tranzitivitate, idempotenți, identitate, involuție, legile lui De Morgan
 - Hedges
 - Variabile fuzzy
 - Definiție
 - Proprietăți
- **Stabilirea variabilelor fuzzy și a mulțimilor fuzzy pe baza funcțiilor de apartenență → mecanism**

Sisteme inteligente – SBC – sisteme fuzzy

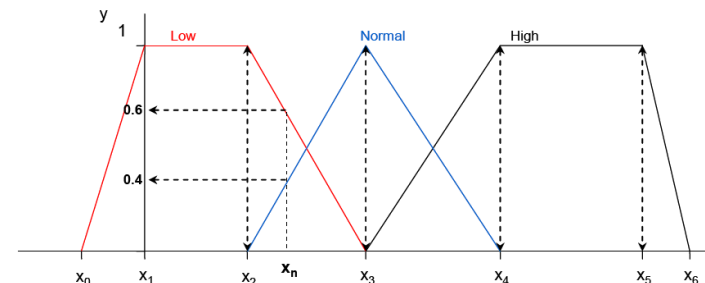
Conținut și arhitectură → fuzzificarea datelor de intrare

□ Mecanism

- Se stabilesc datele (de intrare și ieșire) brute ale sistemului
- Pentru fiecare dată de intrare se definesc funcțiile de apartenență (2, 3 sau mai multe)
 - Fiecărei funcții de apartenență i se asociază o etichetă calitativă – variabilă lingvistică
 - unei variabile brute îi pot corespunde una sau mai multe variabile lingvistice
 - Exemplu
 - Variabilă brută: temperatura T
 - Variabile lingvistice asociate: redusă → A_1 , medie → A_2 , înaltă → A_3
- Se transformă fiecare dată brută de intrare într-o variabilă lingvistică → fuzzificare
 - Se stabilește mulțimea fuzzy din care face parte variabila brută
 - Cum?
 - Pentru o valoare brută dată a unei variabile (care poate aparține uneia sau mai multor mulțimi fuzzy) se calculează valoarea asociată funcției caracteristice corespunzătoare pentru fiecare dintre mulțimile de apartenență (folosind funcția de apartenență)

□ Exemplu

- $T (=x_n) = 5^\circ$
- $A_1 \rightarrow \mu_{A_1}(T) = 0.6$
- $A_2 \rightarrow \mu_{A_2}(T) = 0.4$



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → fuzzificarea datelor de intrare

□ Mecanism

■ Exemplu pentru reglarea unui aparat de aer condiționat

□ Intrări:

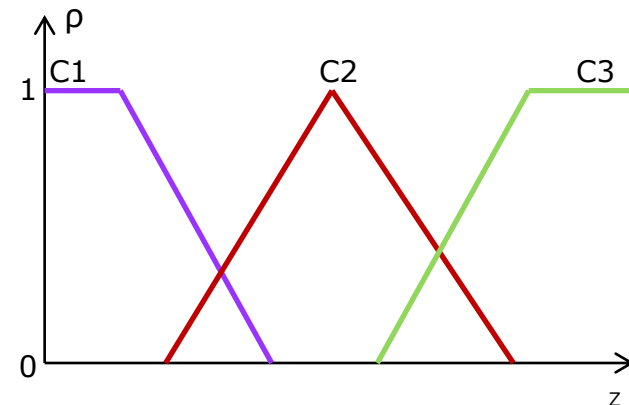
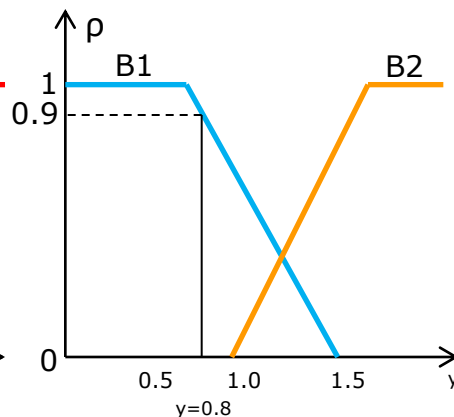
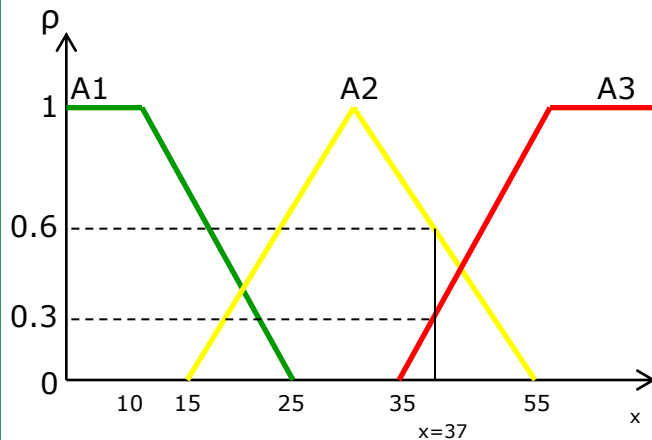
- x (temperatura – rece, normal, fierbinte) și
- y (umiditatea – mică, mare)

□ Ieșiri:

- z (puterea aparatului – redusă, medie, ridicată)

□ Date de intrare:

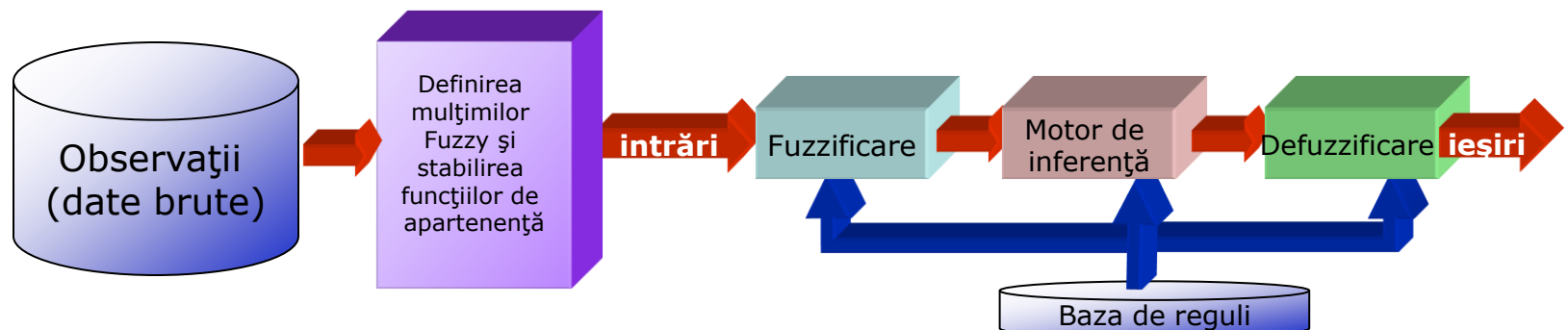
- Temperatura $x = 37$
 - $\mu_{A1}(x)=0$, $\mu_{A2}(x)=0.6$, $\mu_{A3}(x)=0.3$
- Umiditatea $y = 0.8$
 - $\mu_{B1}(x)=0.9$, $\mu_{B2}(x)=0$



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură

- Pași în construirea unui sistem fuzzy
 - Definirea intrărilor și ieșirilor de către expert
 - Datele de intrare și ieșire brute
 - Fuzzificarea datelor de intrare și ieșire
 - Stabilirea variabilelor fuzzy și a mulțimilor fuzzy pe baza funcțiilor de apartenență
 - **Construirea unei baze de reguli de către expert**
 - **Matricea de decizie a bazei de cunoștințe**
 - Evaluarea regulilor
 - Inferența – transformarea intrărilor fuzzy în ieșiri fuzzy prin aplicarea regulilor din baza de cunoștințe
 - Agregarea rezultatelor
 - Defuzificarea
 - Interpretarea rezultatelor



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Construirea unei baze de reguli de către expert

□ Reguli

■ Definiție

□ Construcții lingvistice de tipul

- Afirmațiilor: A
- Condiționărilor: dacă A atunci B

□ Unde A și B sunt (colecții de) propoziții care conțin variabile lingvistice

- A – premisa regulii
- B – consecința regulii

■ Tipologie

□ Necondiționale

- x este (în) A_i
- ex. *Salvează energia*

□ Condiționale

- Dacă x este (în) A_i atunci y este (în) C_k
- Dacă x este (în) A_i și y este (în) B_j atunci z este (în) C_k
- Dacă x este (în) A_i sau y este (în) B_j atunci z este (în) C_k

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Construirea unei baze de reguli de către expert

- Reguli
 - Exemplu

	Reguli în logica clasică	Reguli în logica fuzzy
R_1	<i>Dacă temperatura este -5 atunci vremea este rece</i>	<i>Dacă temperatura este joasă atunci vremea este rece</i>
R_2	<i>Dacă temperatura este 15 atunci vremea este călduță</i>	<i>Dacă temperatura este medie atunci vremea este călduță</i>
R_3	<i>Dacă temperatura este 35 atunci vremea este caniculară</i>	<i>Dacă temperatura este ridicată atunci vremea este caniculară</i>

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Construirea unei baze de reguli de către expert

□ Reguli

■ Baza de reguli fuzzy

- R_{11} : Dacă x este A_1 și y este B_1 atunci z este C_u
- R_{12} : Dacă x este A_1 și y este B_2 atunci z este C_v
- ...
- R_{1n} : Dacă x este A_1 și y este B_n atunci z este C_x

- R_{21} : Dacă x este A_2 și y este B_1 atunci z este C_y
- R_{22} : Dacă x este A_2 și y este B_2 atunci z este C_z
- ...
- R_{2n} : Dacă x este A_2 și y este B_n atunci z este C_v

- ...

- R_{m1} : Dacă x este A_m și y este B_1 atunci z este C_z
- R_{m2} : Dacă x este A_m și y este B_2 atunci z este C_v
- ...
- R_{mn} : Dacă x este A_m și y este B_n atunci z este C_u

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Construirea unei baze de reguli de către expert

□ Reguli

■ Proprietăți ale bazei de reguli

□ Completitudine

- O bază de reguli fuzzy este completă
 - dacă orice valoare de intrare are asociată o valoare între 0 și 1
 - dacă orice variabilă fuzzy este completă
 - dacă mulțimile folosite în bază au suport ne-compact

□ Consistență

- O mulțime de reguli fuzzy este inconsistentă dacă 2 reguli care au aceleași premise conduc la consecințe diferite
 - Dacă x este A_1 și y este B_1 atunci z este C_r
 - Dacă x este A_1 și y este B_1 atunci z este C_v

■ Probleme cu baza de reguli fuzzy

□ Explozia regulilor

- Numărul regulilor crește exponențial odată cu numărul variabilelor de intrare
- Numărul de combinații ale mulțimilor de intrare este
 - unde a i -a variabilă este formată din p_i mulțimi $P = \prod_{i=1}^n p_i$

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Construirea unei baze de reguli de către expert

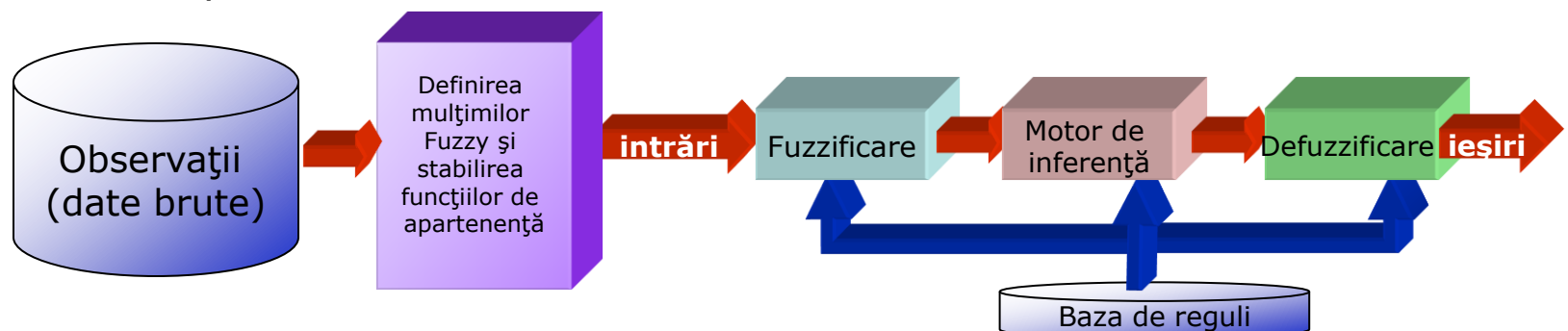
- Matricea de decizie a bazei de cunoștințe
 - Exemplu pentru reglarea unui aparat de aer condiționat
 - Intrări:
 - x (temperatura – rece, normal, fierbinte) și
 - y (umiditatea – mică, mare)
 - Ieșiri:
 - z (puterea aparatului – redusă, menținută constantă, mărită)
 - Reguli:
 - *Dacă temperatura este normală și umiditatea mică atunci puterea aparatului de aer condiționat trebuie să fie menținută constantă*

		Data de intrare y	
		Mică	Mare
Data de intrare x	Rece	Redusă	Constantă
	Normal	Constantă	Mărită
	Fierbinte	Mărită	Mărită

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură

- Pași în construirea unui sistem fuzzy
 - Definirea intrărilor și ieșirilor de către expert
 - Datele de intrare și ieșire brute
 - Fuzzificarea datelor de intrare și ieșire
 - Stabilirea variabilelor fuzzy și a mulțimilor fuzzy pe baza funcțiilor de apartenență
 - Construirea unei baze de reguli de către expert
 - Matricea de decizie a bazei de cunoștințe
 - **Evaluarea regulilor**
 - **Inferența – transformarea intrărilor fuzzy în ieșiri fuzzy prin aplicarea regulilor din baza de cunoștințe**
 - Agregarea rezultatelor
 - Defuzificarea
 - Interpretarea rezultatelor



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Evaluarea regulilor (inferența fuzzy)

□ Care reguli se evaluează mai întâi?

■ Inferență de tip Fuzzy

- Regulile se evaluează în **paralel**, fiecare regulă contribuind la forma rezultatului final.
- Mulțimile fuzzy rezultate sunt defuzzificate **abia după ce toate regulile** au fost evaluate

Amintim

■ Inferență înainte

- Pentru o stare a problemei se colectează informațiile necesare și se aplică regulile posibile pentru BC construită

■ Inferență înapoi

- Se identifică regulile care determină apariția faptului obiectiv și (dacă este posibil) se aplică aceste reguli

□ Cum se evaluează regulile?

- Evaluarea antecedentelor
- Evaluarea consecințelor

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Evaluarea regulilor (inferența fuzzy)

□ Evaluarea antecedentelor

- Pentru fiecare premisă a unei reguli (dacă x este (în) A) se ia în considerare gradul de apartenență al datelor brute la fiecare mulțime fuzzy din care face parte (anterior calculat în etapa de fuzzificare)
- O regulă poate avea una sau mai multe premise (antecedente) legate prin operatori logici: *AND*, *OR* sau *NOT* → Aplicarea operatorilor fuzzy în evaluarea regulilor
 - Operator *AND* → intersecția (minimul) a 2 mulțimi
 - $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$
 - Operator *OR* → reuniunea (maximul) a 2 mulțimi
 - $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$
 - Operator *NOT* → negația (complementul) unei mulțimi
 - $\mu_{\neg A}(x) = 1 - \mu_A(x)$
- Rezultatul evaluării premisei unei reguli
 - Gradul în care antecedentul unei reguli este satisfăcut
 - Poate fi rezultatul unei operații de tip *AND* sau *OR*
 - Alte denumiri:
 - Rule's firing strength
 - Degree of fulfillment

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Evaluarea regulilor (inferența fuzzy)

- Evaluarea consecințelor (aplicarea metodei de implicare)
 - Determinarea rezultatelor
 - stabilirea gradului de apartenență la mulțimile fuzzy corespunzătoare a variabilelor existente în consecința regulilor
 - Fiecare regiune de ieșire va trebui defuzzificată pentru obținerea valorilor crisp
 - În funcție de tipul consecințelor unei reguli
 - Modelul Mamdani – consecința regulii este de forma “variabila de ieșire face parte dintr-o mulțime fuzzy”
 - Modelul Sugeno – consecința regulii este de forma “variabila de ieșire este o funcție crisp care depinde de intrări”
 - Modelul Tsukamoto – consecința regulii este de forma “variabila de ieșire face parte dintr-o mulțime fuzzy cu o funcție de apartenență monotonă”

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Evaluarea regulilor (inferența fuzzy) → evaluarea consecințelor

□ Modelul Mamdani

■ Ideea de bază:

- consecința regulii este de forma “variabila de ieșire face parte dintr-o mulțime fuzzy”
- Rezultatul evaluării premiselor se aplică pentru funcția de apartenență a consecinței
- Exemplu
 - ***Dacă x este în A și y este în B atunci z este în C***

■ Tipologie (în funcție de modul de aplicare a rezultatului asupra funcției de apartenență a consecinței)

- Mulțimi fuzzy rezultat de tip *clipped*
- Mulțimi fuzzy rezultat de tip *scaled*

Sisteme inteligente – SBC – sisteme fuzzy

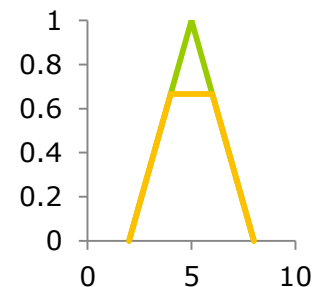
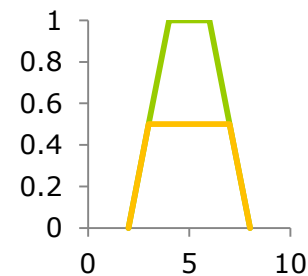
Conținut și arhitectură → Evaluarea regulilor (inferența fuzzy) → evaluarea consecințelor

□ Modelul Mamdani

- Tipologie (în funcție de modul de aplicare a rezultatului asupra funcției de apartenență a consecinței)

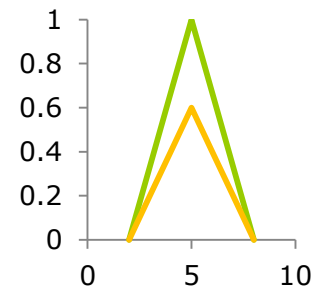
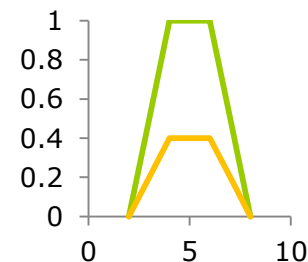
□ Mulțimi fuzzy rezultat de tip *clipped*

- Funcția de apartenență a consecinței se taie la nivelul indicat de valoarea de adevăr a rezultatului
- Avantaj → calcule ușoare
- Dezavantaj → se pierde informații



□ Mulțimi fuzzy rezultat de tip *scaled*

- Funcția de apartenență a consecinței se ajustează prin scalare (multiplicare) valoarea de adevăr a rezultatului
- Avantaj → se pierde mai puțină informație
- Dezavantaj → calcule mai complicate



Sisteme inteligente – SBC – sisteme fuzzy

- Conținut și arhitectură → Evaluarea regulilor → evaluarea consecințelor → Modelul Mamdani
 - Exemplu pentru reglarea unui aparat de aer condiționat
 - Intrări:
 - x (temperatura – rece (A_1), normal (A_2), fierbinte (A_3)) și
 - y (umiditatea – mică (B_1), mare (B_2))
 - Ieșiri:
 - z (puterea aparatului – redusă (C_1), medie (C_2), ridicată (C_3))
 - Date de intrare:
 - Temperatura $x = 37$
 - $\mu_{A_1}(x)=0$, $\mu_{A_2}(x)=0.6$, $\mu_{A_3}(x)=0.3$
 - Umiditatea $y = 0.8$
 - $\mu_{B_1}(x)=0.9$, $\mu_{B_2}(x)=0$

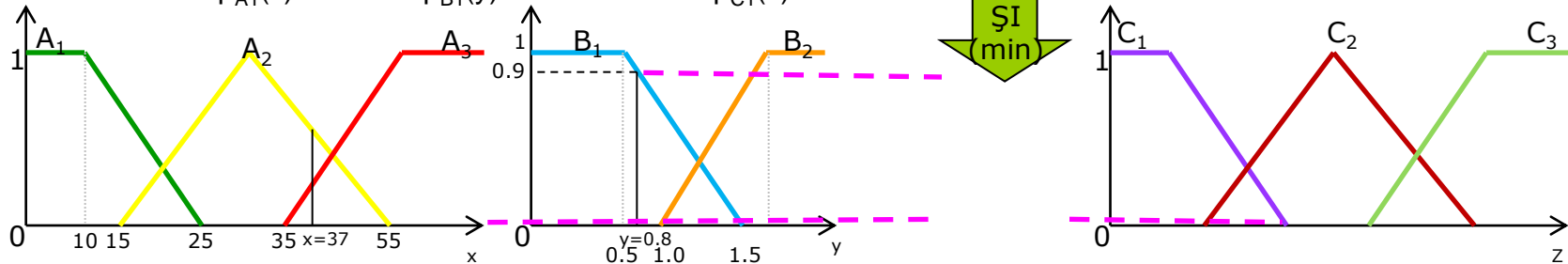
		Data de intrare y	
		Mică	Mare
Data de intrare x	Rece	Redusă	Constantă
	Normal	Constantă	Mărită
	Fierbinte	Mărită	Mărită

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Evaluarea regulilor → evaluarea consecințelor → Modelul Mamdani

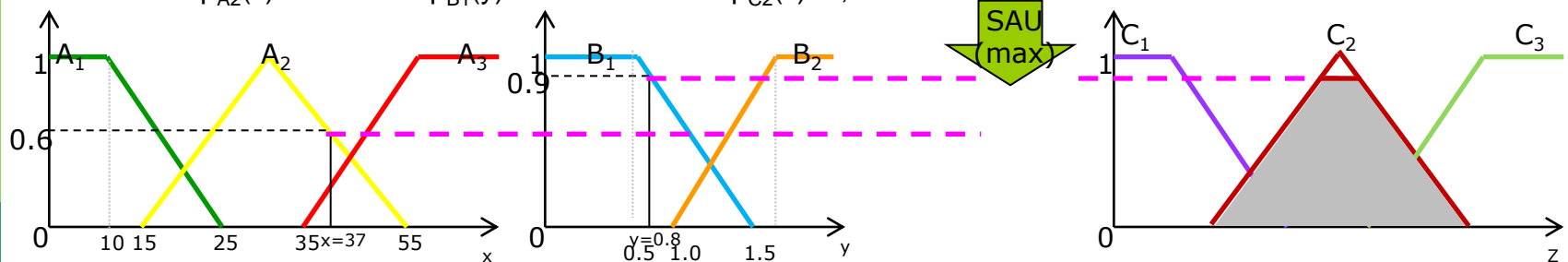
R1: dacă x este în A_1 și y este în B_1 atunci z este în C_1

$\mu_{A_1}(x) = 0$ $\mu_{B_1}(y) = 0.9$ $\mu_{C_1}(z) = 0$



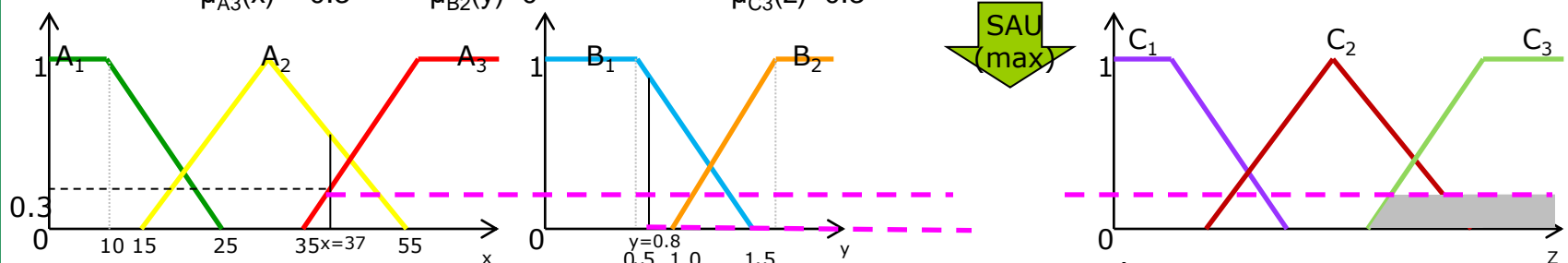
R2: dacă x este în A_2 sau y este în B_1 atunci z este în C_2

$\mu_{A_2}(x) = 0.6$ $\mu_{B_1}(y) = 0.9$ $\mu_{C_2}(z) = 0.9$



R3: dacă x este în A_3 sau y este în B_2 atunci z este în C_3

$\mu_{A_3}(x) = 0.3$ $\mu_{B_2}(y) = 0$ $\mu_{C_3}(z) = 0.3$



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Evaluarea regulilor (inferența fuzzy) → evaluarea consecințelor

□ Modelul Sugeno

■ Ideea de bază

- consecința regulii este de forma “variabila de ieșire este o funcție crisp care depinde de intrări”

□ Exemplu

- **Dacă x este în A și y este în B atunci z este $f(x,y)$**

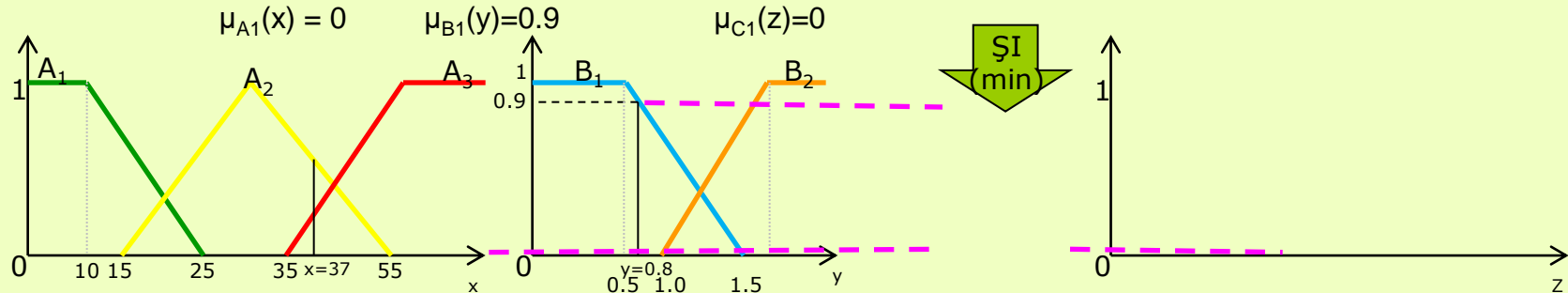
□ Tipologie (în funcție de caracteristicile lui $f(x,y)$)

- Model Sugeno de grad 0 → dacă $f(x,y) = k$ – constantă (funcțiile de apartenență ale consecințelor sunt de forma *singleton* – mulțime fuzzy a cărei funcție de apartenență ia valoarea 1 pentru un singur punct din universul de discurs și 0 în toate celelalte puncte)
- Model Sugeno de grad 1 → dacă $f(x,y) = ax + by + c$

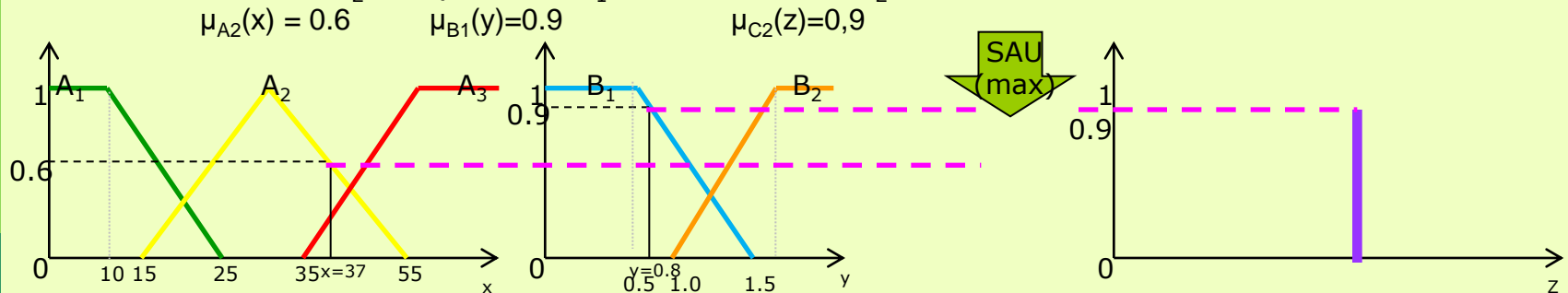
Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Evaluarea regulilor → evaluarea consecințelor → Modelul Sugeno

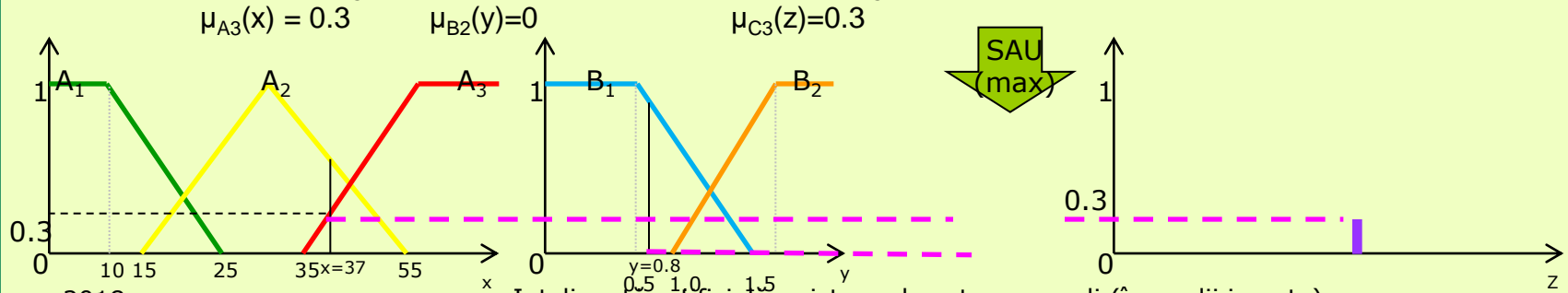
R1: dacă x este în A_1 și y este în B_1 atunci z este în C_1



R2: dacă x este în A_2 sau y este în B_1 atunci z este în C_2



R3: dacă x este în A_3 sau y este în B_2 atunci z este în C_3



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Evaluarea regulilor (inferența fuzzy) → evaluarea consecințelor

□ Modelul Tsukamoto

■ Ideea de bază

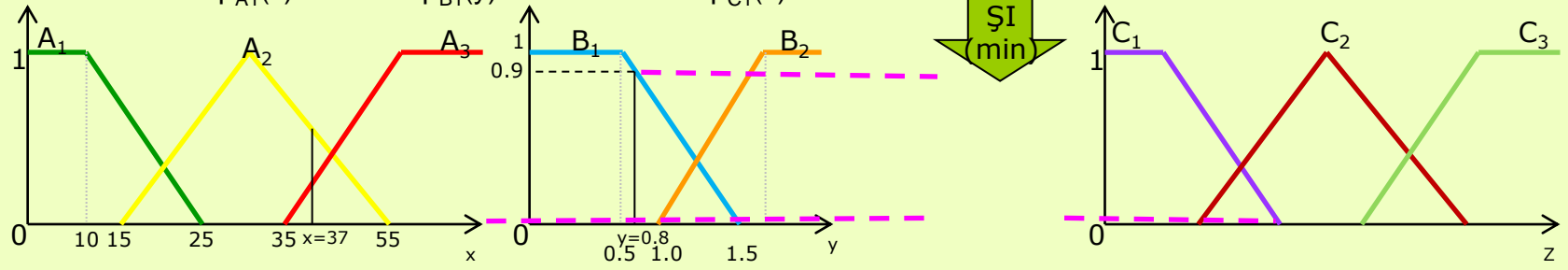
- consecința regulii este de forma “variabila de ieșire face parte dintr-o mulțime fuzzy cu o funcție de apartenență monotonă”
 - Prin inferență se obține o ieșire crisp indusă gradul de satisfacere a regulii (*rule's firing strength*)

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Evaluarea regulilor → evaluarea consecințelor → Modelul Tsukamoto

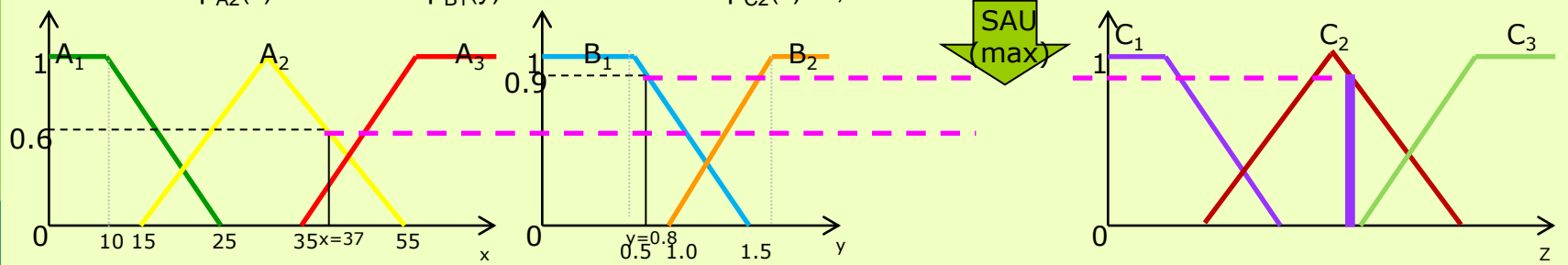
R1: dacă x este în A_1 și y este în B_1 atunci z este în C_1

$\mu_{A_1}(x) = 0$ $\mu_{B_1}(y) = 0.9$ $\mu_{C_1}(z) = 0$



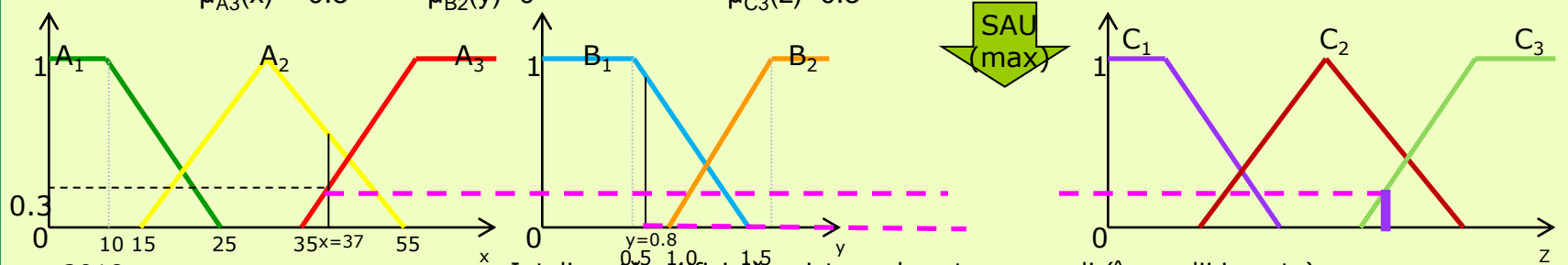
R2: dacă x este în A_2 sau y este în B_1 atunci z este în C_2

$\mu_{A_2}(x) = 0.6$ $\mu_{B_1}(y) = 0.9$ $\mu_{C_2}(z) = 0.9$



R3: dacă x este în A_3 sau y este în B_2 atunci z este în C_3

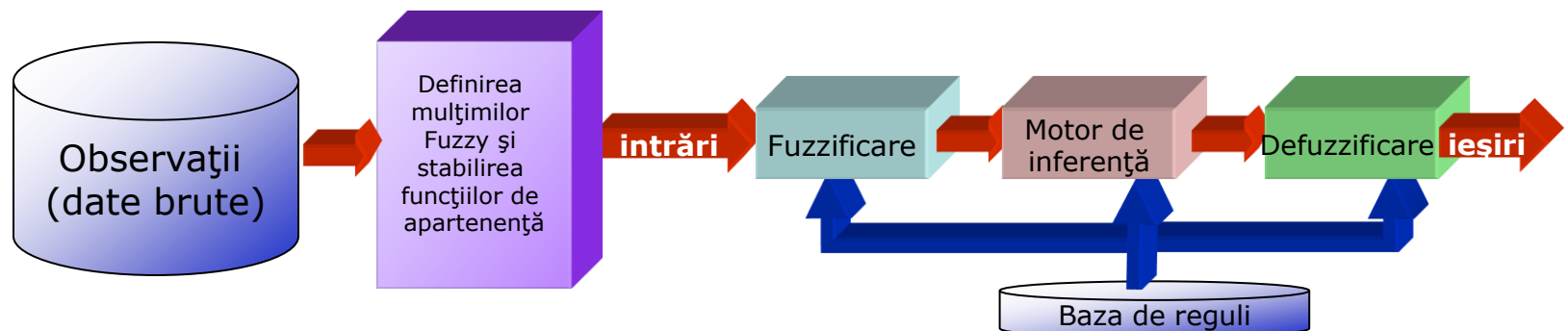
$\mu_{A_3}(x) = 0.3$ $\mu_{B_2}(y) = 0$ $\mu_{C_3}(z) = 0.3$



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură

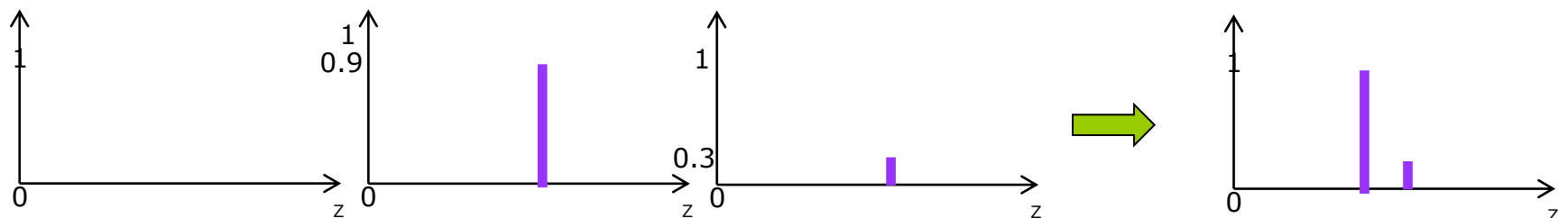
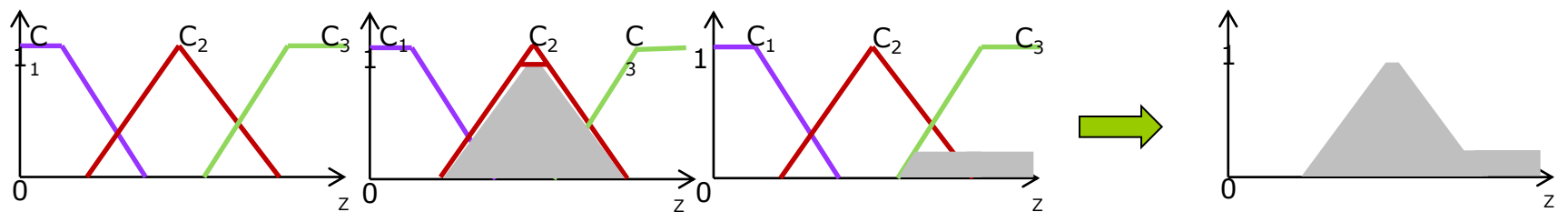
- Pași în construirea unui sistem fuzzy
 - Definirea intrărilor și ieșirilor de către expert
 - Datele de intrare și ieșire brute
 - Fuzzificarea datelor de intrare și ieșire
 - Stabilirea variabilelor fuzzy și a mulțimilor fuzzy pe baza funcțiilor de apartenență
 - Construirea unei baze de reguli de către expert
 - Matricea de decizie a bazei de cunoștințe
 - Evaluarea regulilor
 - Inferența – transformarea intrărilor fuzzy în ieșiri fuzzy prin aplicarea regulilor din baza de cunoștințe
 - **Agregarea rezultatelor**
 - Defuzificarea
 - Interpretarea rezultatelor



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Agregarea rezultatelor

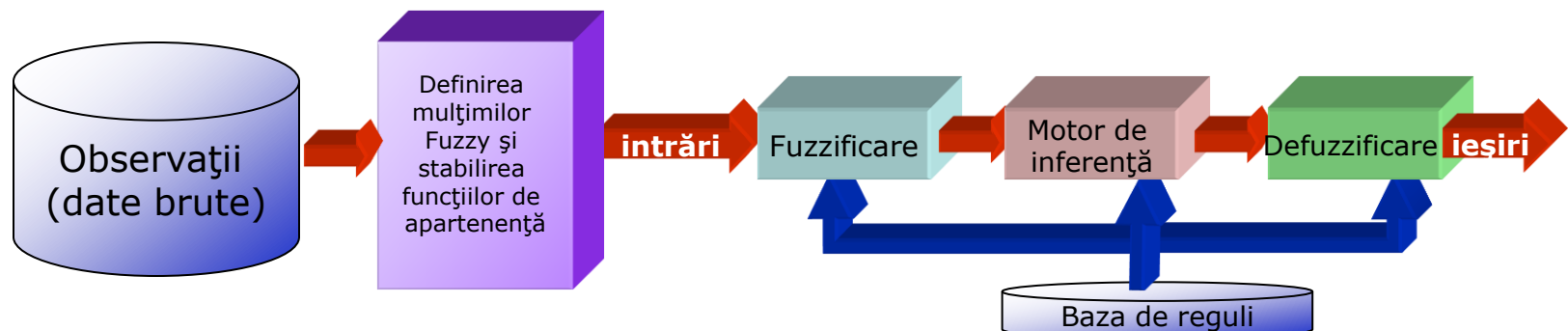
- Unificarea ieșirilor tuturor regulilor aplicate
- Se iau funcțiile de apartenență corespunzătoare tuturor consecințelor și se combină într-o singură mulțime fuzzy, respectiv într-un singur rezultat
- Procesul de agregare are ca
 - intrări → funcțiile de apartenență (tăiate sau scalate) ale consecințelor
 - ieșire → o mulțime fuzzy pentru variabila de ieșire
- Exemplu
 - Mamdani



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură

- Pași în construirea unui sistem fuzzy
 - Definirea intrărilor și ieșirilor de către expert
 - Datele de intrare și ieșire brute
 - Fuzzificarea datelor de intrare și ieșire
 - Stabilirea variabilelor fuzzy și a mulțimilor fuzzy pe baza funcțiilor de apartenență
 - Construirea unei baze de reguli de către expert
 - Matricea de decizie a bazei de cunoștințe
 - Evaluarea regulilor
 - Inferența – transformarea intrărilor fuzzy în ieșiri fuzzy prin aplicarea regulilor din baza de cunoștințe
 - Agregarea rezultatelor
 - **Defuzificarea**
 - Interpretarea rezultatelor



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Defuzificarea

- Ideea de bază
 - Transformarea rezultatului fuzzy al agregării într-o valoare crisp
 - Inferența → obținerea unor regiuni fuzzy pentru fiecare variabilă de ieșire
 - Defuzzificarea → Fiecare regiune de ieșire trebuie defuzzificată pentru a produce valori crisp
 - Se poate pierde informație
- Metode
 - Bazate pe centrul de greutate
 - Centrul de greutate (COG sau COA – *Centroid Area*)
 - Bisectoarea (BOA – *Bisector of area*)
 - Bazate pe maximul funcției de apartenență
 - Media maximelor (MOM - *Mean of maximum*)
 - Cel mai mic maxim (SOM - *Smallest of maximum*)
 - Cel mai mare maxim (LOM - *Largest of maximum*)

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Defuzificarea → Metode

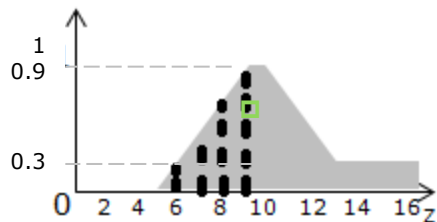
Centrul de greutate (COG sau COA – Centroid Area)

- Găsește punctul z din mijlocul mulțimii agregate

$$COG = \frac{\sum_{i=0}^n x_i \mu_A(x_i)}{\sum_{i=0}^n \mu_A(x_i)} \quad \text{sau} \quad COG = \frac{\int x_i \mu_A(x_i)}{\int \mu_A(x_i)}$$

Exemplu

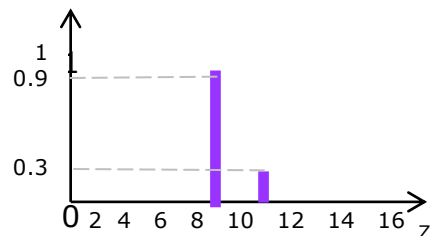
- Modelul Mamdani → estimarea centrului prin calcularea COG pe baza unui eșantion de n puncte ($x_i, i = 1, 2, \dots, n$) din mulțimea fuzzy rezultat



$$COG = \frac{5*0 + 6*0.3 + 7*0.5 + 8*0.7 + 9*0.9 + 10*0.9 + 11*0.7 + 12*0.5 + 13*0.3 + 14*0.3 + 15*0.3 + 16*0.3}{0 + 0.3 + 0.5 + 0.7 + 0.9 + 0.9 + 0.7 + 0.5 + 0.3 + 0.3 + 0.3 + 0.3}$$

$$COG \cong 13.7$$

- Modelul Sugeno sau Tsukamoto, COG devine media ponderată a celor m valori crisp rezultate prin aplicarea tuturor regulilor (în număr de m)



$$COG = \frac{9*0.9 + 11*0.3}{0.9 + 0.3}$$

$$COG \cong 9.5$$

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Defuzificarea → Metode

□ Bisectoarea (BOA – Bisector of area)

- Găsește punctul z în care o linie verticală va separa mulțimea agregată în 2 părți de arii egale

$$BOA = \int_{\alpha}^z \mu_A(x) dx = \int_z^{\beta} \mu_A(x) dx,$$

unde $\alpha = \min\{x \mid x \in A\}$ și $\beta = \max\{x \mid x \in A\}$

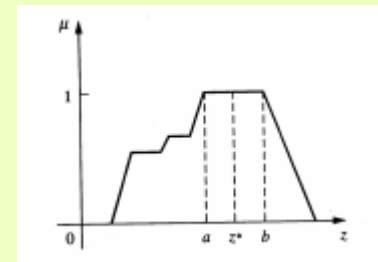
Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Defuzificarea → Metode

□ Media maximelor (MOM - *Mean of maximum*)

- Găsește punctul z care reprezintă media punctelor din mulțimea agregată care au valoarea funcției de apartenență (μ) maximă

$$MOM = \frac{\sum_{x_i \in \max \mu} x_i}{|\max \mu|}, \text{ unde } \max \mu = \mu^* = \{x \mid x \in A, \mu(x) = \max\}$$



□ Cel mai mic maxim (SOM - *Smallest of maximum*)

- Găsește cel mai mic punct z din mulțimea agregată care are valoarea funcției de apartenență (μ) maximă

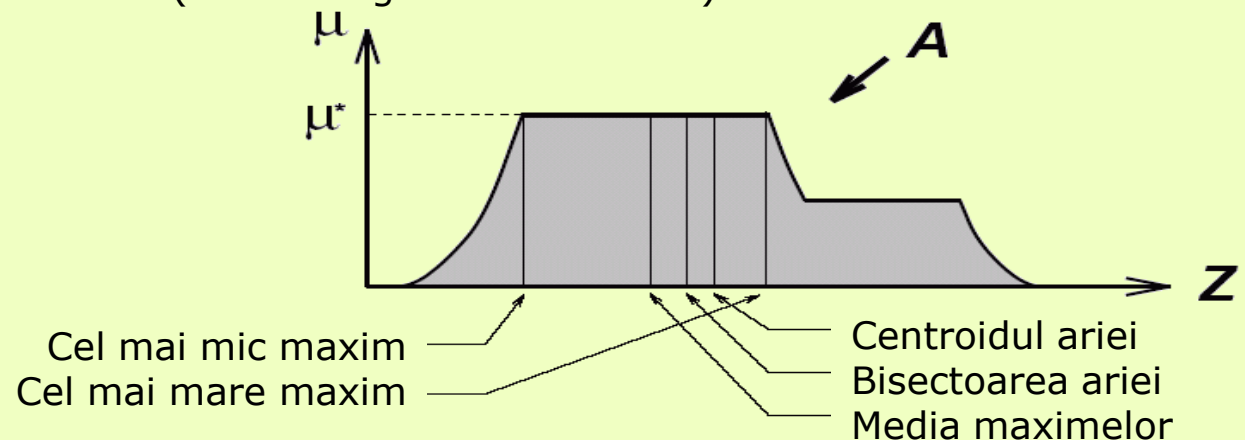
□ Cel mai mare maxim (LOM - *Largest of maximum*)

- Găsește cel mai mare punct z din mulțimea agregată care are valoarea funcției de apartenență (μ) maximă

Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură → Defuzificarea

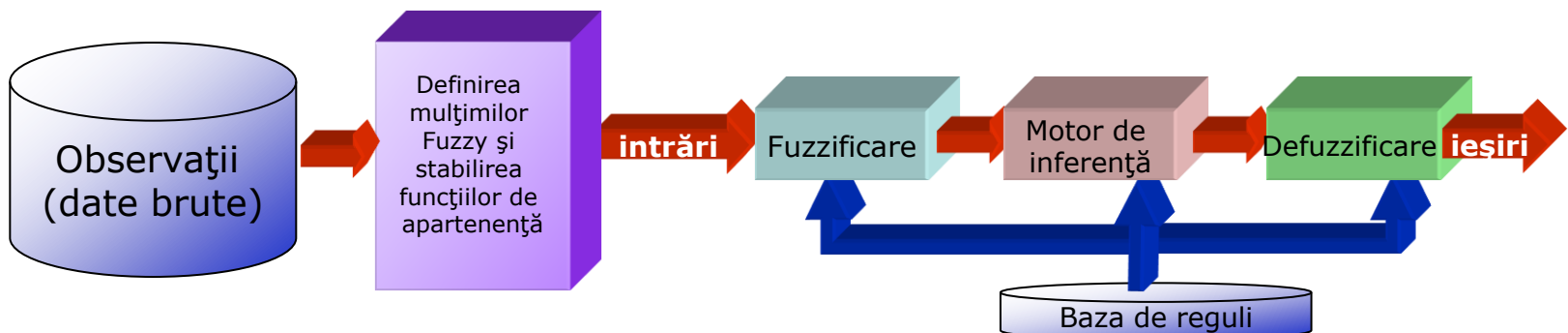
- Ideea de bază
 - Transformarea rezultatului fuzzy al agregării într-o valoare crisp
- Metode
 - Bazate pe centrul de greutate
 - Centrul de greutate (COG sau COA – Centroid Area)
 - Bisectoarea (BOA – Bisector of area)
 - Bazate pe maximul funcției de apartenență
 - Media maximelor (MOM - Mean of maximum)
 - Cel mai mic maxim (SOM - Smallest of maximum)
 - Cel mai mare maxim (LOM - Largest of maximum)



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură

- Pași în construirea unui sistem fuzzy
 - Definirea intrărilor și ieșirilor de către expert
 - Datele de intrare și ieșire brute
 - Fuzzificarea datelor de intrare și ieșire
 - Stabilirea variabilelor fuzzy și a mulțimilor fuzzy pe baza funcțiilor de apartenență
 - Construirea unei baze de reguli de către expert
 - Matricea de decizie a bazei de cunoștințe
 - Evaluarea regulilor
 - Inferența – transformarea intrărilor fuzzy în ieșiri fuzzy prin aplicarea regulilor din baza de cunoștințe
 - Agregarea rezultatelor
 - Defuzificarea
 - **Interpretarea rezultatelor**



Sisteme inteligente – SBC – sisteme fuzzy

Conținut și arhitectură

□ Pași în construirea unui sistem fuzzy

■ Definirea intrărilor și ieșirilor de către expert

- Datele de intrare și ieșire brute
- Fuzzificarea datelor de intrare și ieșire

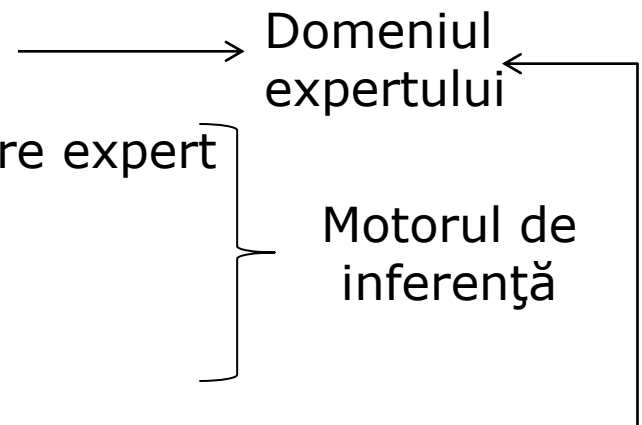
■ Construirea unei baze de reguli de către expert

■ Evaluarea regulilor

■ Agregarea rezultatelor

■ Defuzificarea

■ Interpretarea rezultatelor



Sisteme inteligente – SBC – sisteme fuzzy

□ Avantaje

- Se pot folosi reguli pentru exprimarea impreciziei și aproximărilor din lumea reală
- Ușor de înțeles, testat și întreținut
- Robuste → pot opera și când regulile nu sunt foarte clare
- Necesită mai puține reguli decât alte SBC
- Regulile se evaluează în paralel

□ Dezavantaje

- Necesită multe simulări și testări
- Nu învață singure
- Este dificilă stabilirea celor mai corecte reguli
- Lipsa unui model matematic exact

Sisteme inteligente – SBC – sisteme fuzzy

Aplicații

- Control în spațiu
 - Alitudinea sateliților
 - Reglaje în avioane
- Control auto
 - Transmisie automată, controlul traficului, Sisteme împotriva blocării roților în timpul frânării
 - *Dacă frâna este caldă și viteza nu este foarte mare atunci presiunea de frânare se reduce ușor*
- Business
 - Sisteme de decizie, evaluarea personalului, managementul fondurilor, predicții de piață, sisteme de asigurarea a securității tranzacțiilor
- Industrie
 - Controlul schimbului de energie, controlul purificării apei,
 - Controlul pH-ului, distilarea chimică, producerea de polimeri, formarea metalelor
- Electronică
 - Controlul expunerii automate a camerelor foto/video, controlul umidității, sisteme de aer condiționat, Sisteme pentru reglajul dușului
 - Sisteme pentru reglarea congelatoarelor
 - Sisteme pentru reglarea mașinilor de spălat (încărcătura, concentrația de detergent, etc.)

Sisteme inteligente – SBC – sisteme fuzzy

Aplicații

- Alimentație
 - Procese de producere a brânzei
- Militar
 - Recunoașterea subacvatică, recunoașterea imaginilor în infraroșu, sisteme de decizie în traficul naval
- Marină
 - Piloți automați pentru nave, și submarine selecția automată a rutelor
- Medical
 - Sisteme de diagnosticare (diabet, cancer), controlul presiunii arteriale în timpul anesteziei, modelarea rezultatelor neuropatologice pentru pacienții cu Alzheimer
- Robotică
 - Controlul manipulatorilor flexibili și a brațelor robotice

Recapitulare



□ SBC

- Sisteme computaționale în care baza de cunoștințe și modulul de control se suprapun

□ SBC pot funcționa

■ În medii certe

- SBL
- SBR

■ În medii incerte

- Sisteme de tip Bayes
 - Regulile au asociate probabilități de realizare
- Sisteme bazate pe factori de certitudine
 - Faptele și regulile au asociați factori de certitudine
- Sisteme fuzzy
 - Faptele au asociate grade de apartenență la anumite mulțimi

□ Informațiile prezentate au fost colectate din diferite surse de pe internet, precum și din cursurile de inteligență artificială ținute în anii anteriori de către:

■ Conf. Dr. Mihai Oltean –
www.cs.ubbcluj.ro/~moltean

■ Lect. Dr. Crina Groșan -
www.cs.ubbcluj.ro/~cgrosan

■ Prof. Dr. Horia F. Pop -
www.cs.ubbcluj.ro/~hfpop