

## 1. Sisteme „Expert”

Firma de recrutare (Partea a 2-a). Va angajati la o firma de recrutare resurse umane. Aici vi se spune sa identificati oameni care studiaza mult. Pentru a va face munca mai usoara, construiti un Sistem Expert folosind urmatoarele reguli:

R0: **if** ?x are absente **then** ?x are restante

R1: **if** AND(?x e inteligent; ?x e harnic) **then** ?x are note bune

R2: **if** OR(?x are note bune; ?x are restante) **then** ?x studiaza mult

R3: **if** OR(?x nu pierde vremea; ?x e eficient) **then** ?x a harnic

Dupa cateva interviuri apar urmatoarele afirmatii despre candidati:

A0: Ioana are absente;

A1: Maria e eficienta;

A2: Maria e inteligenta

Folosind Backward chaining demonstrati ca **Maria studiaza mult; Ioana studiaza mult.**

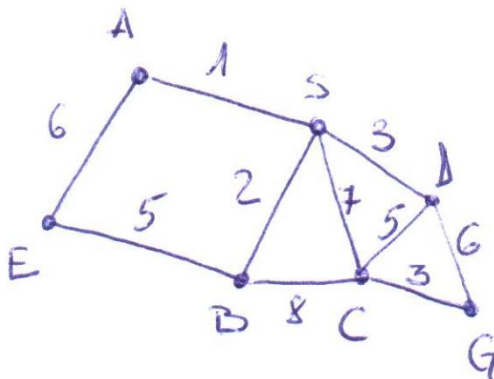
Folosind Forward chaining, rutati primele 3 iteratii pentru Sistemul Expert.

## 2. Algoritmi de cautare

Mini Facedook (+Dragobete). Va hotarati sa va ajutati prietenii ca sa gaseasca persoana iubita. Pentru aceasta folositi Mini-Facedook, o baza de date cu persoane. In acest sens, folositi un graf si modelati persoanele ca noduri si legaturile dintre ele ca arce. Pentru a gasi o legatura intre prieten si persoana iubita implementati un algoritm de cautare. Folositi prima data Cautarea in adancime (Depth-First Search), apoi folositi Cautarea pe Nivel (BFS). Ce ruta apare intre S si G?

Vreti ca persoanele sa se intalneasca cat ma repede. Pe noduri e inscriptionat timpul de discutie pentru ca persoana respectiva sa ofere ajutorul (timpul de convingere). Folositi Branch&Bound pentru a stabili ruta cea mai rapida de cunoastere intre S si G.

Observati ca algoritmul nu este foarte eficient si ca treceti de mai multe ori prin acelasi loc in cautare. Va decideti sa folositi B&B + Lista elementelor extinse (B&B + EL). Care este arborele de cautare? Exista imbunatatire fata de B&B ?

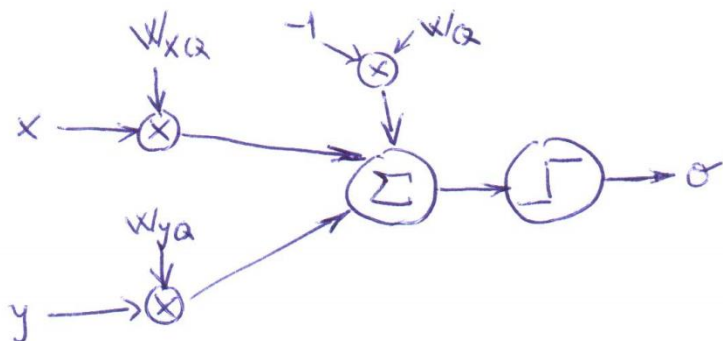


Minimax. Va plictisiti acasa, asa ca va decideti sa construiti un agent inteligent care sa va fie adversar in jocuri. Agentul gandeste cateva mutari in avans dupa care produce evaluari statice, astfel rezulta urmatorul arbore de joc. Aplicand algoritmul minimax, determinati valoarea din varf a maximizatorului si secventa de mutari care produce acest rezultat. Tineti cont ca factorul de ramificare este 2, si valorile statice sunt urmatoarele:

21 -84 -2 46 -5 66 8 -89 95 47 -48 -5 45 67 93 -19

### 3. Algoritmi de clasificare. Retele neuronale

In cadrul experimentelor de inteligenta artificiala primiti la studiu un neuron artificial. Pentru neuronul din figura, cu o functie de activare tip treapta si cu ponderile  $W_{QX} = 1$ ;  $W_{QY} = 1$ ;  $W_Q = 1$ , demonstrati desenand limita de decizie ca este capabil sa clasifice corect populatia formata din elemente din clasa 1: (1,1); (2,0); (2,2) si clasa 2: (-2,0); (-2,-1); (0,2). Va ganditi ca este util sa reprezentati grafic cele 6 elemente intr-un sistem xOy. Calculati iesirea din neuron pentru fiecare element. Apare un element nou (-1,1), cum se clasifica acesta? Ce trebuie facut ca elementul nou sa fie clasificat ca apartinand clasei 1?



$$\text{treapta}(x) = \begin{cases} 0, & \text{dacă } x < 0 \\ 1, & \text{dacă } x \geq 0 \end{cases}$$

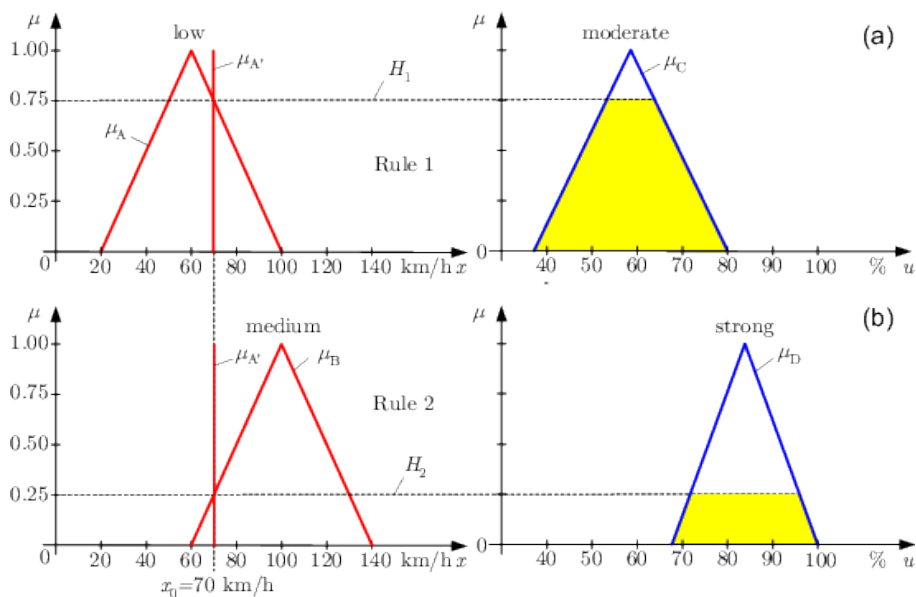
### 4. Fuzzy logic

Se dau doua reguli fuzzy. Functiile de apartenenta corespunzatoare sunt prezentate in figura de mai jos.

(1) IF the car speed is low THEN the brake force is moderate

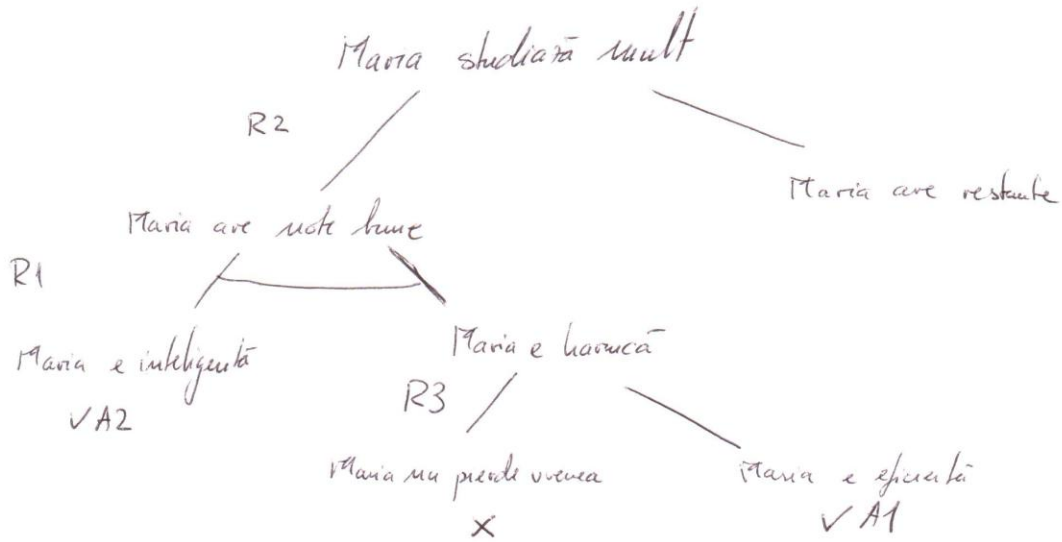
(2) IF the car speed is medium THEN the brake force is strong

Calculati valoarea procentuala a franarii, rezultata in urma procesului de defuzzyficare, tinand cont ca viteza de deplasare este de 70 km/h

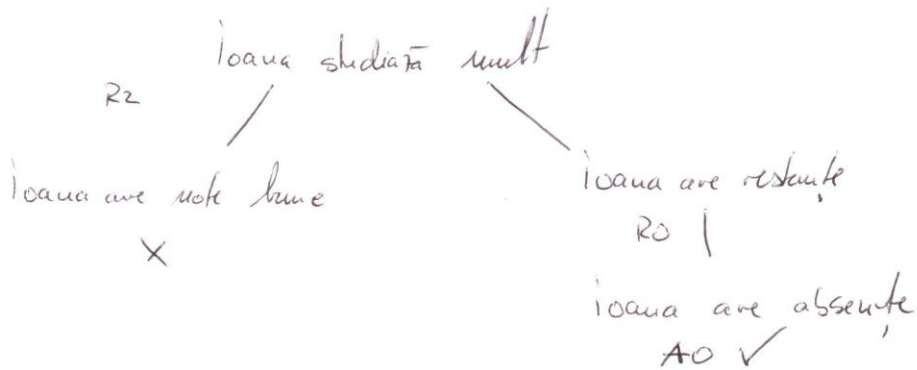


1. RBFS.

Backward Chaining.



⇒ „Maria studiază mult” se achievează



⇒ „Ioana studiază mult” se achievează

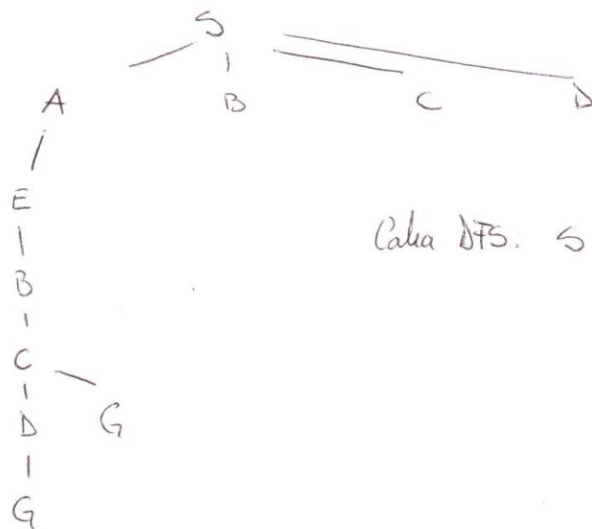
# 1. RBES

## Forward Chaining

Iterația	se poate executa	Se execută	Afirmative nouă	Explică
1.	R0, R3	R0	A3. ioana are restanțe	R0 - ioana
2.	R0, R3	R3	A4. Maria e harnică	R3 - Maria
3.	R0, R1, R3	R1	A5. Maria are voie bună	R1 - Maria

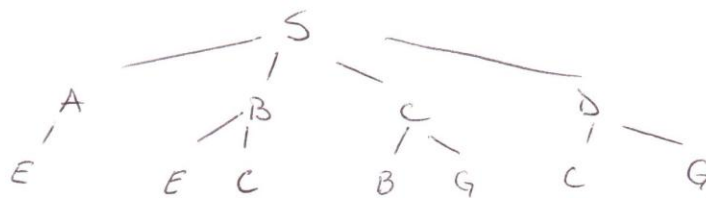
## 2. Algoritmi de căutare

DFS



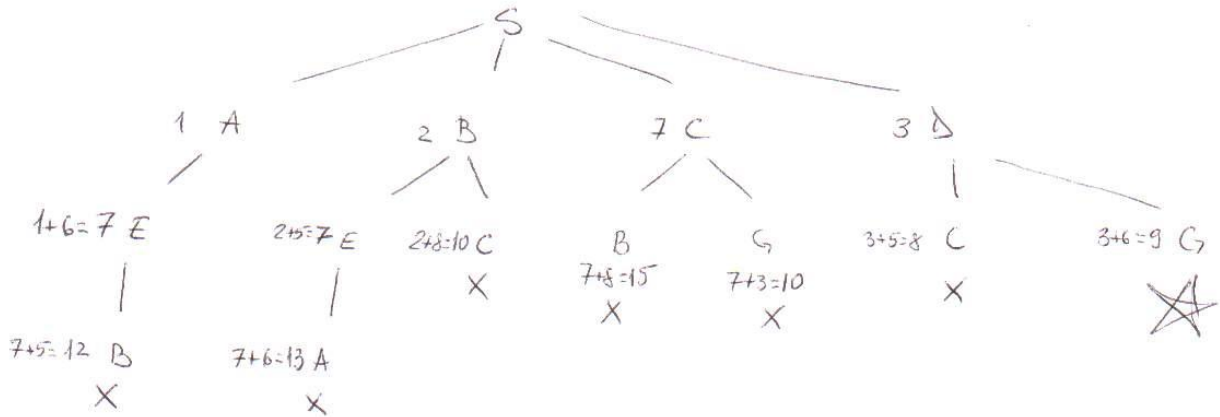
Calea DFS: S A E B C D G

BFS



Calea BFS: S C G

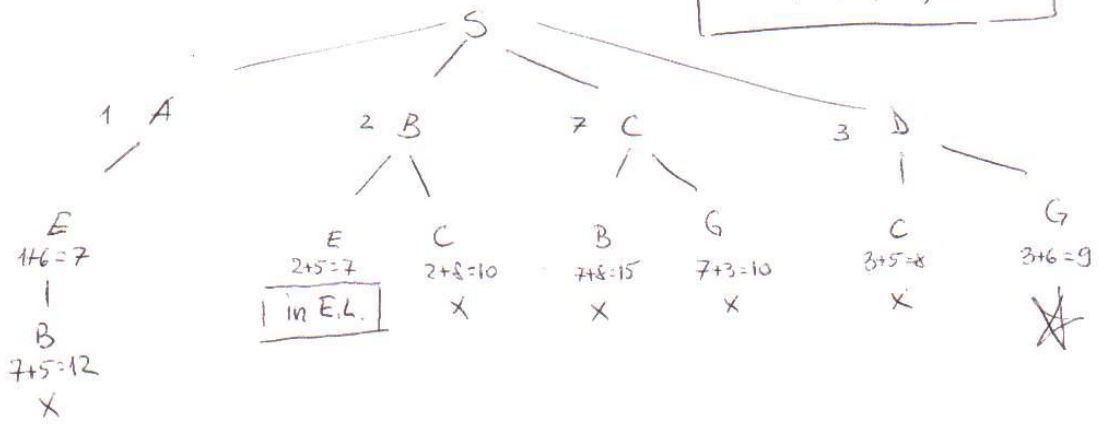
B&B



Calea găsită prin B&B = S D G  
Lungimea = 9

B&B + E.L.

Elemente in  
Extended List:  
A, B, D, C, E

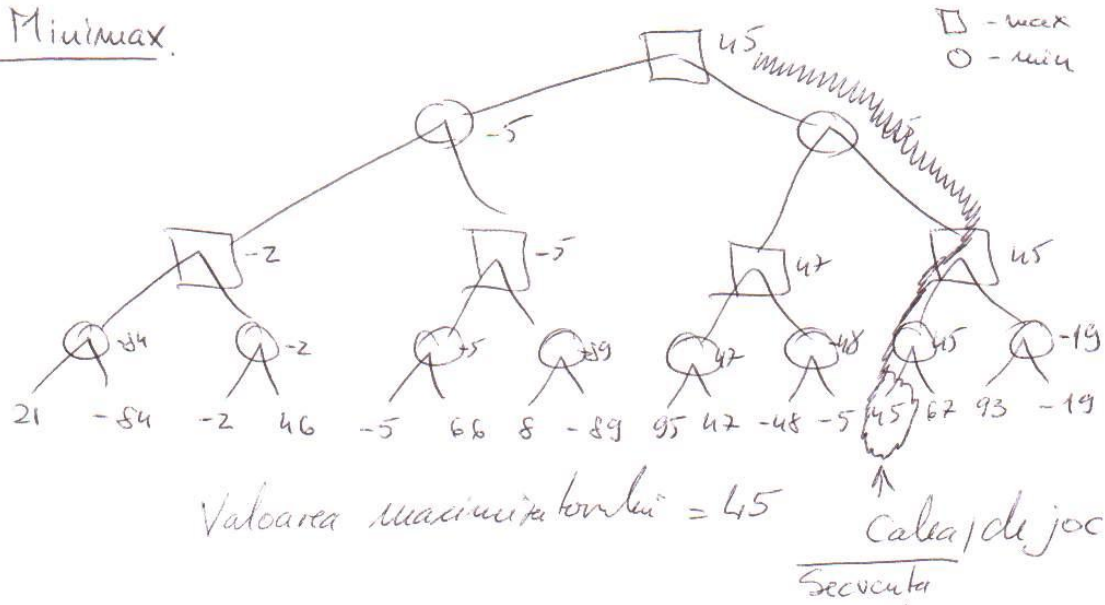


Calea găsită prin B&B = S D G  
Lungimea = 9

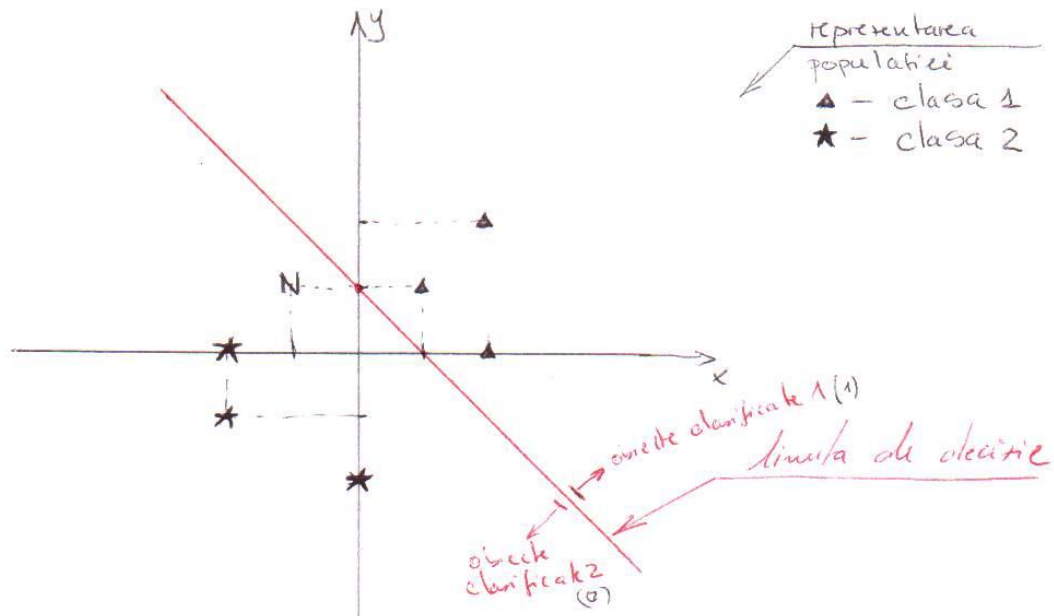
Există îmbunătățiri, în sensul că, pe calea SBE, nodul E nu se mai extinde, el fiind deja în Extended List. => economie de timp și calcule.

Notă Se notază cu X ramurile care nu se mai extind (de exemplu pt. că sunt mai lungi decât calea la final)

## 2. Minimax.



## 3. Algoritmi de clasificare. Rețele Neuronale Artificiale.



Pentru a determina linia de decizie, scriem ecuația neuronului.

$$x \cdot w_{x2} + y \cdot w_{y2} - 1 \cdot w_2 \leq 0 \quad (\text{rezultatul acestei intrări în funcția treaptă de activare } \sigma \text{ produce } 1 \text{ sau } 0)$$

$$\Downarrow$$

$$y \cdot w_{y2} = w_2 - x \cdot w_{x2}$$

$$\Downarrow$$

$$y = \frac{w_2 - x \cdot w_{x2}}{w_{y2}} ; \text{ dacă } x \text{ introduce valori numerice } \Rightarrow$$

$$\boxed{y = 1 - x} = \text{ecuația dreptei care produce linia de decizie}$$

Testarea din neuron:

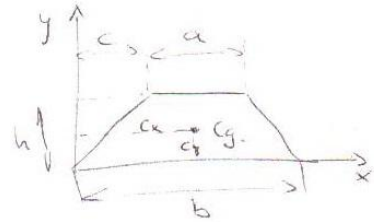
x	y		
(1, 1)	$1 \cdot 1 + 1 \cdot 1 - 1 \cdot 1 = 1$	$\text{treapta}(1) = 1$	} 1 pentru clasa 1
(2, 0)	$2 \cdot 1 + 0 \cdot 1 - 1 \cdot 1 = 1$	$\text{treapta}(1) = 1$	
(2, 2)	$2 \cdot 1 + 2 \cdot 1 - 1 \cdot 1 = 3$	$\text{treapta}(3) = 1$	
↑ clasa 1 ↓ clasa 2			
(-2, 0)	$-2 \cdot 1 + 0 \cdot 1 - 1 \cdot 1 = -3$	$\text{treapta}(-3) = 0$	} 0 pt clasa 2
(-2, -1)	$-2 \cdot 1 + -1 \cdot 1 - 1 \cdot 1 = -4$	$\text{treapta}(-4) = 0$	
(0, -2)	$0 \cdot 1 + -2 \cdot 1 - 1 \cdot 1 = -3$	$\text{treapta}(-3) = 0$	
Element nou:			
(-1, 1)	$-1 \cdot 1 + 1 \cdot 1 - 1 \cdot 1 = -1$	$\text{treapta}(-1) = 0$	⇒ elementul nou e clasat in clasa 2.

Obiectul nou  $N(-1, 1)$  este clasificat ca facăni d park din  
clasa 2.

Pentru clasificarea obiectului nou in clasa 1 trebuie ajustat  
ponderile neuronului ~~prin~~. Acest proces se realizează prin  
învățare — algoritmul Backpropagation.

## 4. Fuzzy Logic

C.g. al unui trapez.  $\frac{2ac + a^2 + cb + ab + b^2}{3(a+b)}$   
(pe direcția x)



In cazul nostru  $a=0$  (e vorba de triunghi)

$$C_{gT1} = \frac{20 \cdot 40 + 40^2}{3 \cdot 40} = \frac{500 + 1600}{120} = 20 + \text{offset} = 60 \cdot 0,75 = 45$$

$$C_{gT2} = \frac{15 \cdot 30 + 30^2}{3 \cdot 30} = \frac{450 + 900}{90} = 55 + \text{offset} = 85 \cdot 0,25 = 21,25$$

$$C_{g \text{ figură}} = \frac{60 \cdot 0,75 + 85 \cdot 0,25}{0,75 + 0,25} = 61,25$$