

BLIA-MAS Laborator 07

Cuprins:

- analogie între ingineria cunostintelor și programare
- exemplul circuitului electronic
- lumea supermarketului

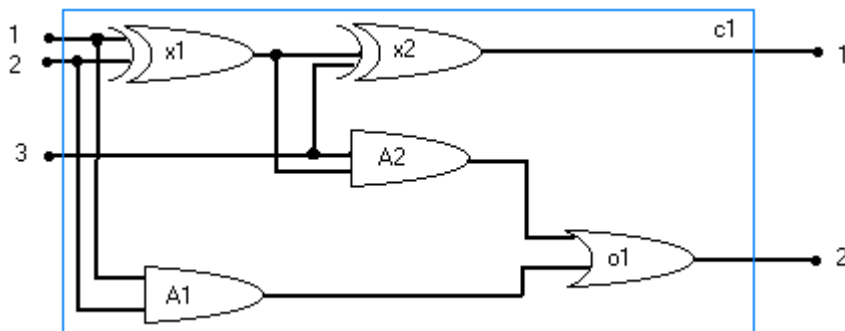
Analogie între ingineria cunostintelor și programare

<i>Ingineria cunostintelor</i>	<i>Programarea</i>
(1) Alegerea unei logici	Alegerea unui limbaj de programare
(2) Construirea unei baze de cunostinte	Scrierea programului
(3) Implementarea unei teorii valide	Alegerea sau scrierea unui compilator
(4) Introducerea de noi fapte	Rularea programului

În cazul programului, ieșirea este derivată din intrare și program. Pentru o bază de cunostinte, răspunsurile sunt derivate din descrierea problemelor și din baza de cunostinte.

Avantajul ingineriei cunostintelor este că necesită mai puțin angajament, deci mai puțină muncă. Un inginer al cunostintelor trebuie doar să decidă ce obiecte și relații merită să fie reprezentate, și ce relații tin de ce obiecte. Un programator trebuie să facă toate acestea, plus trebuie să decidă cum să compute relațiile dintre obiecte, atunci când se dau niște date de intrare. Un inginer al cunostintelor specifică ce este adevărat, și procedura de inferență trebuie să își dea seama cum să determine faptul să devină o soluție a problemei. Deoarece un fapt este adevărat indiferent de ce sarcină are cineva de rezolvat, baza de cunostinte, în principiu, poate fi reutilizată la rezolvarea diferitelor sarcini, fără a necesita modificarea. Repararea unei baze de cunostinte este foarte ușor deoarece orice propoziție este adevărată sau falsă prin ea însăși, pe când corectitudinea unui program depinde foarte mult de contextul lui.

Exemplul circuitului electronic



(Russel, J., Norvik, P. - *Artificial Intelligence A modern Approach*)

Scopul este acela de a furniza o analiza asupra circuitului si de a determina daca circuitul este de fapt un sumator , si daca se pot raspunde la intrebari referitor la valorile in diferite puncte ale circuitului.

Circuitele digitale sunt compuse din sarme si porti. Semnalele curg de-a lungul sarmelor catre intrarile portilor, si fiecare poarta produce un semnal de iesire care este transmis la randul lui de alta sarma. Avem patru tipuri de porti: AND, OR si XOR care au exact doua intrari si portile NOT care au o singura intrare. Toate portile au o singura iesire.

Intai de toate trebuie sa distingem portile una de alta. Intai de toate vom numi portile cu niste constante: $X1$, $X2$ etc. Apoi vom preciza tipul lor: $Tip(X1)=XOR$. Apoi consideram terminalele. O poarta poate avea unul sau mai multe terminale de intrare si unul sau mai multe terminale de iesire. Le vom numerota cu niste constante. Poarta $X1$ va avea terminalele $X1Intrare1$, $X2Intrare2$, $X1Iesire1$. In locul notatiilor destul de anevoioase, vom folosi functii (de exemplu in loc de $X1Iesire1$, vom folosi $Iesire(1,X1)$). Analog se va folosi functia $Intrare$. Modul de conectare va fi un predicat $Conecteaza$ care avea doua argumente ($Conecteaza(Iesirea(1,X1), Intrarea(1,X2))$). In final vom dori sa aflam daca un semnal este *Activ* sau *Inactiv*. Vom mai introduce o functie *Semnal* care preia ca argument un terminal si intoarce valoarea semnalului.

Codarea regulilor generale

1. Daca doua terminale sunt conectate, atunci ele au acelasi semnal:
 $\forall t_1, t_2 \text{ Conecteaza}(t_1, t_2) \Rightarrow \text{Semnal}(t_1) = \text{Semnal}(t_2)$
2. Semnalul de la fiecare terminal este sau Activ sau Inactiv (dar nu ambele):
 $\forall t \text{ Semnal}(t) = \text{Activ} \vee \text{Semnal}(t) = \text{Inactiv}$
 $\text{Activ} \neq \text{Inactiv}$
3. Conectarea este un predicat comutativ:
 $\forall t_1, t_2 \text{ Conecteaza}(t_1, t_2) \Leftrightarrow \text{Conecteaza}(t_2, t_1)$
4. Iesirea unei porti OR este activa daca si numai daca intrarile ei sunt active:
 $\forall g \text{ Tip}(g) = \text{OR} \Rightarrow \text{Semnal}(Iesire(1, g)) = \text{Activ} \Leftrightarrow \exists n \text{ Semnal}(Intrare(n, g)) = \text{Activ}$
5. Iesirea unei porti AND este inactiva daca si numai daca orice intrare a ei este inactiva:
 $\forall g \text{ Tip}(g) = \text{AND} \Rightarrow \text{Semnal}(Iesire(1, g)) = \text{Inactiv} \Leftrightarrow \exists n \text{ Semnal}(Intrare(n, g)) = \text{Inactiv}$
6. Iesirea unei porti XOR este activa daca si numai daca intrarile ei sunt diferite:
 $\forall g \text{ Tip}(g) = \text{XOR} \Rightarrow \text{Semnal}(Iesire(1, g)) = \text{Activ} \Leftrightarrow \text{Semnal}(Intrare(1, g)) \neq \text{Semnal}(Intrare(2, g))$
7. Iesirea unei porti NOT este diferita de intrarea sa:
 $\forall g \text{ Tip}(g) = \text{NOT} \Rightarrow \text{Semnal}(Iesire(1, g)) \neq \text{Semnal}(Intrare(1, g))$

Codarea instantelor specifice

$Tip(X1)=XOR$ $Tip(X2)=XOR$
 $Tip(A1)=AND$ $Tip(A2)=AND$
 $Tip(O1)=OR$

<i>Conecteaza(Iesire(1, X1), Intrare(1, X2))</i>	<i>Conecteaza(Iesire(1, C1), Intrare(1, X1))</i>
<i>Conecteaza(Iesire(1, X1), Intrare(2, A2))</i>	<i>Conecteaza(Iesire(1, C1), Intrare(1, A1))</i>
<i>Conecteaza(Iesire(1, A2), Intrare(1, O1))</i>	<i>Conecteaza(Iesire(2, C1), Intrare(2, X1))</i>
<i>Conecteaza(Iesire(1, A1), Intrare(2, O1))</i>	<i>Conecteaza(Iesire(2, C1), Intrare(2, A1))</i>
<i>Conecteaza(Iesire(1, X2), Intrare(1, C1))</i>	<i>Conecteaza(Iesire(3, C1), Intrare(2, X2))</i>
<i>Conecteaza(Iesire(1, O1), Intrare(2, C1))</i>	<i>Conecteaza(Iesire(3, C1), Intrare(1, A2))</i>

Intrebari pentru procedura de inferenta

Ce combinatie de intrari va cauza prima iesire a lui C₁ sa fie inactiva si a doua iesire a lui C₁ sa fie activa?

$\exists i_1, i_2, i_3$ *Semnal(Intrare(1, C₁))=i₁ \wedge Semnal(Intrare(2, C₁))=i₂ \wedge
 *Semnal(Intrare(3, C₁))=i₃ \wedge
 *Semnal(Iesire(1, C₁))=Inactiv \wedge Semnal(Iesire(2, C₁))=Activ***

Raspunsul va fi:

*(i₁ = Activ \wedge i₂ = Activ \wedge i₃ = Inactiv) \vee
 (i₁ = Activ \wedge i₂ = Inactiv \wedge i₃ = Activ) \vee
 (i₁ = Inactiv \wedge i₂ = Activ \wedge i₃ = Activ)*

Intrebare:

Care sunt posibilele seturi de valori ale tuturor terminalelor pentru circuitul de adunare?

$\exists i_1, i_2, i_3, o_1, o_2$ *Semnal(Intrare(1, C₁))=i₁ \wedge Semnal(Intrare(2, C₁))=i₂ \wedge
 *Semnal(Intrare(3, C₁))=i₃ \wedge
 *Semnal(Iesire(1, C₁))=o₁ \wedge Semnal(Iesire(2, C₁))=o₂***

Lumea supermarketului

(Russel, J., Norvik, P. - Artificial Intelligence A modern Approach)

In acest exemplu va trebui sa definim baza de cunostinte de care are nevoie agentul pentru a face cumparaturi dintr-un supermarket.

Percepte:

1. Agentul primeste trei percepte la fiecare moment de timp: simtire, sunet, si vizualizare.

2. Perceptul de simtire este asemanator cu cel descris in lumea wumpusului. Agentul precepe o busitura doar daca in momentul anterior de timp se executa actiunea *Inainte* si nu avem destul loc in casuta in care doreste sa se mute.
3. Sunetul perceput este o lista de cuvinte vorbite. Agentul precepe cuvintele vorbite de agenti pe o raza de doua celule.
4. Daca aparatul agentului este fixata pe detalii, atunci el va precepe detaliat imaginile vizuale ale fiecarui obiect pe o raza de trei celule (inainte si pe diagonala).
5. Daca aparatul agentului nu este setata pe detalii, agentul va precepe imaginile obiectelor, pe o raza de trei celule (inainte si pe diagonala), destul de brut.
6. Un percept vizual consta dintr-o locatie relativa, o dimensiune aproximativa, culoare, forma, si posibil alte caracteristici.

Actiuni:

1. Un agent poate vorbi un set de cuvinte.
2. Un agent se poate duce o celula inainte.
3. Un agent se poate roti 90° catre stanga sau catre dreapta.
4. Un agent isi poate focaliza aparatul inspre celula curenta, sau oricare celula pe o raza de 3 ceule in fata sau pe diagonala in fata.
5. Un agent isi poate focaliza aparatul dinspre celula curenta, sau oricare celula pe o raza de 3 ceule in fata sau pe diagonala in fata.
6. Un agent poate apuca un obiect care se afla pe o raza de o ceula in jurul lui. Pentru a realiza acest lucru trebuie sa specifice coordonatele exacte ale obiectului, si ii mai trebuie o mana libera.
7. Un agent poate lasa un obiect pe care-l are in mana. Pentru a face acest lucru trebuie sa specifice coordonatele relative ale punctului unde doreste sa elibereze obiectul.

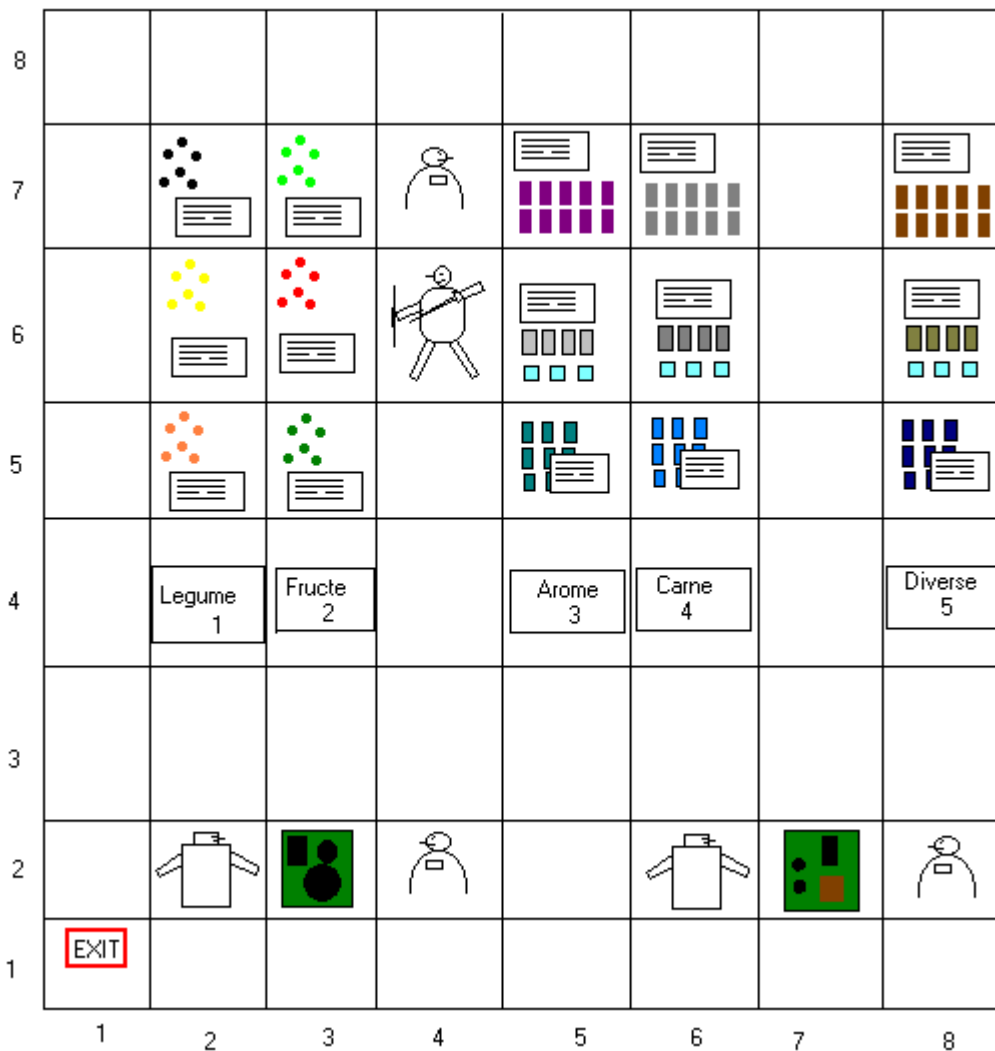
Scopul initial al agentului este de a cumpara toate produsele de pe lista de cumparaturi. Scopul poate fi modificat daca unele produse nu exista sau sunt prea scumpe. Agentul trebuie sa faca cumparaturile destul de repede si fara a se ciocni de prea multe ori de diferite obiecte. Un scop mai ambitios ar fi sa ii dam agentului scopul de a pregati cina, lasandu-i posibilitatea sa constituie el lista de cumparaturi.

Mediul este reprezentat de interiorul magazinului, impreuna cu celelate obiecte si oameni. Magazinul este reprezentat de o multime de celule dispuse intr-o matrice. La captul de final al magazinului se afla casele pentru plati si functionarii. Alti cumparatori sau angajati ai magazinului se pot afla oriunde in magazin. Agentul porneste de la intrarea in supermarket si tot acolo trebuie sa si ajunga cand termina de cumparat. Tot acolo se alfa si un semn IESIRE in caz ca agentul uita. De asemenea in magazin avem si diferite semne care simbolizeaza numele raioanelor.

Organizarea cunostintelor

- Planificarea meniului: agentul va trebui sa stie cum sa modifice lista de cumparaturi cand in magazin nu se mai gaseste un anumit produs.
- Navigarea: ca si in lumea wumpusului, agentul va trebui sa inteleaga efectul actiunilor de miscare si sa creeze o harta interna a lumii.

- Adunarea: agentul va trebui sa fie in stare sa gaseasca si sa adune produsele de pe lista. Parte a acestui lucru reprezinta indicerea obiectelor din perchepe: agentul va avea nevoie de reguli de recunoastere pe care sa infereze ca un obiect solid sferic rosu de aproximativ 7 centimetrii ar putea reprezenta o rosie.
- Comunicarea: Agentul va trebui sa fie in stare sa intrebe atunci cand a ajuns intr-o situatie pe care nu poate s-o rezolve.
- Plata: Agentul va trebui sa stie ca 50\$ este prea mult pentru o rosie, si daca pretul total este de 17.35\$ atunci ar trebui sa primeasca restul de 2.65\$ cand schimba o bancnota de 20\$.



Agentul este la [4,5], alti cumparatori la [2,2] si [6,2], casierii la [4,2] si [8,2]. Alti angajati ai magazinului se pot observa la [4,7].

Planificarea meniului

Agentul trebuie sa cumpere ingredientele pentru a organiza o masa. Pentru aceasta isi intocmeste o lista de produse. Atunci cand analizeaza lista, apar doua probleme. Prima, dintr-o lista de produse, agentul trebuie sa induca obiectul compozit pe care aceste produse il constituie. A doua, agentul ar trebui sa decida cum trebuie inlocuite produsele neaccesibile pentru a completa obiectul compozit (in cazul nostru masa de pranz). Acest lucru se poate realiza la doua niveluri: inlocuirea unui ingredient cu altul pentru a completa reteta, si daca acesta nu este posibil, inlocuirea intregii liste (de fapt a componentelor care realizeaza un fel de mancare) cu alta pentru a realiza in final ceva din care sa-si gateasca pranzul.

Primul pas este de a converti lista de cumparaturi (lista de cuvinte) intr-o lista de parti (lista de categorii).

Refera("rosii", Rosii)

Refera("ceapa", Ceapa)

Al doilea pas este de a descrie obiectele in termeni necesari si optionali.

$\forall r, w \text{ ParteNecesara}(r, w) \Rightarrow \forall p p \in r \Rightarrow \text{ParteNecesara}(p, w)$

$\forall o, w \text{ ParteOptionala}(o, w) \Rightarrow \forall p p \in o \Rightarrow \text{ParteNecesara}(p, w)$

$\forall r, w \text{ ParteNecesara}(r, w) \Leftrightarrow \forall c c \in w \Rightarrow \exists i i \in r \wedge \text{ParteDin}(i, c)$

$\forall o, w \text{ ParteOptionala}(o, w) \Leftrightarrow \exists c c \in w \Rightarrow \forall i i \in o \wedge \text{ParteDin}(o, c)$

Urmatorul pas il reprezinta descrierea mancarii si a partilor constituente:

ParteNecesara({FelPrincipal}, Mancare)

ParteOptionala({PrimulFel, Farfurie, Salata, Desert ...}, Mancare)

ParteNecesara({Salata, Rasfirata}, SalataVerde)

ParteOptionala({Rosii, Castraveti, Piper, Morcovi...}, SalataVerde)

ParteNecesara({Pasta, SosBolognese}, PastaBolognese)

ParteOptionala({BrazaTopita}, PastaBolognese)

ParteNecesara({Ceapa, UleiDeMasline, Unt, Morcovi, CarneMacra, Sare, VinAlb, Lapte, Rosii}, SosBolognese)

Apoi avem nevoie de informatii aditionale referitor la ce punem in farfurie:

SalataVerde \subset Salata

Salata \subset Farfurie

PastaBolognese \subset PrimulFel

PrimulFel \subset Farfurie

Rosii \subset SosRosii

RosiiConservate \subset SosRosii

Tagliatelle \subset Pasta

Introducem functia *PotFace*.

$\forall l, d \text{ PotFace}(l, d) \Leftrightarrow d \in \text{Farfurie} \wedge \text{ParteNecesara}(p, d) \wedge \text{FaceParte}(l, \text{Marfa})$
 $\{\text{Sare, Unt, UleiDeMasline}\} \subset \text{Marfa}$

Navigare

Sa zicem ca pe agent il intereseaza rosiile. O problema ar fi localizarea rosiilor. Urmatoarea strategie se poate adopta cu orarecare succes:

1. Daca agentul stie locatia rosiilor dintr-o vizita anterioara, este calculat un drum din locatia curenta pana in locul unde sunt rosiile.
2. Altfel, daca locatia raionului de legume este cunoscuta, se calculeaza un drum pana acolo.
3. Altfel, agentul se misca prin magazin drept inainte pana cand intalneste un indicator pentru raionul de legume.
4. Daca nici una dintre acestea nu merge, agentul se intreaba si cauta pe cineva pe care sa-l intrebe unde-s rosiile (\Rightarrow Comunicarea).
5. Daca agentul gaseste raionul de vegetale, atunci se va plimba de-a lungul raionului cu camera nesetata pe detalii, cautand ceva rosu. Atunci cand intalneste ceva rosu, seteaza camera pe detalii si testeaza daca sunt rosii (\Rightarrow Adunarea).

Adunarea

Agentul va folosi urmatoarele reguli:

1. Daca avem o categorie care se potriveste perceptului, se va presupune ca obiectul este membru al acelei categorii.
2. Daca perceptul se potriveste catorva categorii, dar exista un indicator in apropiere care identifica una dintre ele, se va presupune ca obiectul este mebru al acelei categorii.
3. Daca exista un indicator de raion care indica o categorie (sau o spercategorie), se va presupune ca obiectul apartine acelei categorii.

$\forall x x \in \text{Rosii} \Rightarrow \text{CuloareaSuprafetei}(x, \text{Rosie})$

$\forall x x \in \text{Portocale} \Rightarrow \text{CuloareaSuprafetei}(x, \text{Portocalie})$

$\forall x x \in \text{Mere} \Rightarrow \text{CuloareaSuprafetei}(x, \text{Rosie}) \vee \text{CuloareaSuprafetei}(x, \text{Verde})$

$\forall x x \in \text{Rosii} \Rightarrow \text{Forma}(x, \text{Rotunda})$

$\forall x x \in \text{Portocale} \Rightarrow \text{Forma}(x, \text{Rotunda})$

.....
 $\forall x \text{CuloareaSuprafetei}(x, c) \wedge \text{Visibil}(x) \Rightarrow \text{CauzeazaPerceptCuloare}(x, c)$

$\forall x \text{Forma}(x, s) \wedge \text{Visibil}(x) \Rightarrow \text{CauzeazaPerceptForma}(x, s)$

Comunicarea

Comunicarea printr-un limbaj vorbit este destul de greu. Vom presupune ca agentul comunica prin semne. Daca un cuvânt apare pe un semn/indicator de raion, atunci membrii categoriei pe care o desemneaza cuvântul, vor fi localizati in raionul respectiv.

$$\forall a (a \in \text{Raioane} \wedge \exists s, w \text{ Semn}(s, a) \wedge w \in \text{Cuvinte}(s)) \Rightarrow \\ \exists x, c \text{ Referinta}(w, c) \wedge x \in c \wedge \text{La}(x, a)$$

Daca un cuvânt apare pe un indicator mic, membrii acelei categorii vor fi localizati in apropiere.

$$\forall s, w, l (s \in \text{Semn} \wedge \text{Dimensiune}(s) < \text{Metrii}(0.3) \wedge w \in \text{Cuvinte}(s) \wedge \text{La}(s, l)) \Rightarrow \\ \exists x, c \text{ Referinta}(w, c) \wedge x \in c \wedge \text{La}(x, \text{ZonaDinApropiere}(l))$$

Plata

Agentul trebuie sa stie cand nu trebuie sa plateasca prea mult pentru un produs. Pentru inceput trebuie sa cunoasca preturile rezonabile ale produselor:

$$\forall g \in \text{Tipic}(\text{CarneMacra}) \wedge \text{Greutate}(g) = \text{Kile}(1) \Rightarrow \$ (1) \leq \text{PretRezonabil}(g) \leq \$ (2)$$

Agentul ar trebui sa stie ca pretul total este proportional cu cantitatea, dar de obicei se dau discounturi daca se cumpara cantitati mari.

$$\forall q, c, w, p \ q \in c \wedge \text{Greutate}(q) = w \wedge \text{Pret}(q) = p \Rightarrow \\ \forall m, q_2 \ m > 1 \wedge q_2 \in c \wedge \text{Greutate}(q_2) = m \times w \Rightarrow \\ (1 + (m-1)/2) \times p \leq \text{PretRezonabil}(q_2) \leq m \times p$$

Agentul trebuie sa stie ca este o afacere proasta sa plateasca mai mult pe un produs decat pretul mediu, si ca orice ce cumpara care este o afacere proasta, reprezinta o actiune proasta.

$$\forall i \ \text{Pret}(i) > \text{PretRezonabil}(i) \Rightarrow \text{AfacereProasta}(i) \\ \forall i \ \text{AfacereProasta}(i) \Rightarrow \forall a \ \text{Rau}(\text{Cumpara}(a, i))$$

Agentul trebuie sa cunoasca ca este rau sa paraseasca magazinul, atata timp cat mai are inca ceva in mana care apartine magazinului.

$$\forall a, x, s, i \ s \in \text{Magazin} \wedge T(\text{Cara}(a, x) \wedge \text{La}(x, s) \wedge \text{Apartine}(s, x)) \Rightarrow T(\text{Rau}(\text{Exit}(a)), i)$$

Pentru plata, agentul trebuie sa stie ca trebuie sa se duca la casierie, sa astepte la coada, sa puna produsele cumparate pe banda, sa astepte casierul sa le inregistreze si sa plateasca.

$$\forall b, m, s, p, e \ e \in \text{CumparaDinSupermarket}(b, m, s, p) \Rightarrow \\ \exists e_1, e_2, e_3, e_4, e_5 \quad e_1 = \text{Mergi}(b, c) \wedge \text{VerificaRand}(c) \wedge \\ e_2 = \text{Pune}(b, m, c) \wedge e_3 = \text{PretTotal}(s, m) \wedge \\ e_4 = \text{Pune}(b, p, c) \wedge e_5 = \text{Apuca}(b, m) \wedge \\ \text{Inainte}(e_1, e_2) \wedge \text{Inainte}(e_2, e_3) \wedge \text{Inainte}(e_3, e_4) \wedge \text{Inainte}(e_4, e_5) \wedge \\ \text{ParteDin}(e_1, e) \wedge \text{ParteDin}(e_2, e) \wedge \text{ParteDin}(e_3, e) \wedge \text{ParteDin}(e_4, e) \wedge \text{ParteDin}(e_5, e)$$