

Artificial intelligence - Proiect II  
- Satisfacerea constrangerilor -  
-Logica propozitionala si a predicatelor-

Șandru Diana - Teodora  
Andreicuț Mădălina - Ioana

7/12/2020

# 1 BedTime puzzle

## 1.1 Descriere puzzle

Patru copii, fiecare cu o vârstă specifică, se culcă la ore diferite. Se urmărește KB-ul (clues) pentru a afla vârsta, numele și ora de culcare a fiecărui copil.



### CLUES:

1. Copilul cu vârsta cea mai mare merge la somn cu 30 minute înainte de Harold.
2. Donald se culca mai târziu decât Vincent.
3. Donald are vârsta de 6 ani.
4. Cel mai tânăr băiat se culca la ora 8:00 PM.
5. Copilul care se culca la 8:30 PM este copilul care are vârsta de 6 ani sau Harold.

## 1.2 Tabel de adevăr

Name	Age	Time
Donald	6 years	8:30 PM
Harold	7 years	9:30 PM
Larry	8 years	9:00 PM
Vincent	5 years	8:00 PM

## 1.3 Cod

### 1.3.1 Modelare KB

Donald(a).  
Harold(b).  
Larry(c).

Vincent(d).

Age8(x) & Harold(y)  $\rightarrow$  (Time8PM(x) & Time8halfPM(y) ) |  
 (Time8halfPM(x) & Time9PM(y)) |  
 (Time9PM(x) & Time9halfPM(y)).

Donald(x) & Vincent(y)  $\rightarrow$  (Time8halfPM(x) & Time8PM(y) ) |  
 (Time9PM(x) & Time8halfPM(y)) |  
 (Time9PM(x) & Time8PM(y)) |  
 (Time9halfPM(x) & Time9PM(y)) |  
 (Time9halfPM(x) & Time8halfPM(y)) |  
 (Time9halfPM(x) & Time8PM(y)).

Donald(x)  $\rightarrow$  Age6(x).

Age5(x)  $\rightarrow$  Time8PM(x).

Time8halfPM(x)  $\rightarrow$  (Harold(x) &  $\neg$ Age6(x) ) | ( $\neg$ Harold(x) & Age6(x)).

### 1.3.2 Implementare si testare incrementală

Implementarea si testarea incrementală s-a realizat prin testarea treptată in Prover9 a unor tinte(goal-uri) in ordinea in care ele s-au putut deduce.

Age6(a).

Time8halfPM(a).

Age6(a) & Age7(b) & Age8(c) & Age5(d).

Time8halfPM(a) & Time9halfPM(b) & Time9PM(c) & Time8PM(d).

Time9PM(c).

Time8PM(d).

In primul rand, Donald are 6 ani. Pe urma din ultimul clue, putem alege copilul care adoarme la 8:30 PM ca fiind Donald. De aici mergem cu aceasta premiza pana la finalul puzzle-ului, asta in cazul in care nu intampinam o contradictie. Altfel, vom merge inapoi si vom alege pe copilul care adoarme la 8:30 PM ca fiind Harold. Neintampinand aceasta problema, premiza este una corecta.

## 1.4 Pareri personale

Initial s-a incercat implementarea acestui puzzle cu ajutorul unor functii: earlier(x, y) si later(x, y), care verifica daca x adoarme mai devreme cu 30 minute decat y, respectiv daca x adoarme mai tarziu decat y. Pentru modelarea acestei implemetari am considerat ora culcarii pe pozitii consecutive:

Time8PM(a).

Time8halfPM(b).

Time9PM(c).

Time9halfPM(d).

## 2 Vocation puzzle

### 2.1 Descriere puzzle

Trei studenți au absolvit universitatea pentru a-și urma vocația aleasă - medic, avocat și om de știință. Din aceste informații și următoarele indicii, pentru fiecare student, puteți stabili ce universitate a urmat fiecare, vârsta absolvirii și vocația?



#### CLUES:

1. Cel mai tanar student nu a absolvit ca avocat, nici nu a urmat Universitatea. B
2. Avocatul nu a urmat nici Universitatea A, nici Universitatea B.
3. Peter, care nu a urmat Universitatea A, este cel mai in varsta dintre ceilalti studenti.
4. Studenta (singura de genul feminin dintre studentii in cauza) este mai mare ca varsta decat omul de stiinta si este mai mica decat studentul care urmeaza Universitatea C.
5. Studentul care urmeaza Universitatea C nu a absolvit ca om de stiinta.

### 2.2 Tabel de adevar

Name	University	Vocation	Graduation Age
Peter	University C	Lawyer	26
Paul	University A	Scientist	23
Mary	University B	Doctor	24

### 2.3 Cod

#### 2.3.1 Modelare KB

Peter(a).  
Paul(b).  
Mary(c).

Age23(x)  $\rightarrow$   $\neg$ Lawyer(x) &  $\neg$ UniversityB(x).  
Lawyer(x)  $\rightarrow$   $\neg$ UniversityA(x) &  $\neg$ UniversityB(x).

```

Peter(x) -> -UniversityA(x) & Age26(x).
Mary(x) -> Age24(x).
Scientist(x) -> Age23(x).
UniversityC(x) -> Age26(x).
UniversityC(x) -> -Scientist(x).

```

### 2.3.2 Implementare si testare incrementală

Implementarea si testarea incrementală s-a realizat prin testarea treptată în Prover9 a unor tinte(goal-uri) în ordinea în care ele s-au putut deduce.

```

Age26(a).
Age24(c).
Age23(b).
Scientist(b).
Doctor(c).
Lawyer(a).
.

```

În primul rând, deducem că Peter are 26 de ani, iar Mary are 24 din implicațiile clue-urilor. Pe urmă, verificăm dacă Paul este om de știință, tot din implicațiile clue-urilor. Și așa mai departe până când se deduce întregul tabel de adevăr.

## 2.4 Pareri personale

Acest tip de puzzle de nivel mediu a ajutat la întărirea abilităților de a lucra și modela astfel de tipuri de probleme. Totodată, abordarea incrementală impune un parcurs lent, dar sigur, garantând corectitudinea rezolvării puzzle-ului până la pasul la care s-a ajuns cu testarea.

## 3 Christmas presents puzzle

### 3.1 Descriere puzzle

Moș Crăciun se pregătește de Crăciun - sortând cadouri, umplând sacii și pregătindu-și traseul pentru a se asigura că poate livra toate cadourile sale minunate copiilor lumii la timp, pentru când se vor trezi în dimineața de Crăciun. Problema este că, în fiecare an, chiar înainte de Crăciun, renii nu au ce să facă și, mai ales, se plictisesc. Întotdeauna se întreabă dacă pot ajuta, dar nu pot face nimic până când Moșul nu este gata să plece. Anul acesta, Moșul este pregătit pentru ei și le-a pus o mică provocare pentru a încerca să-i țină ocupați. El le-a dat următoarea problemă: cinci copii din cinci familii diferite care trăiesc în cinci regiuni diferite ale Angliei au cerut câte un cadou diferit, dintre care două sunt o bicicletă și un ponei. Din aceste informații și următoarele indicii, pentru fiecare copil, puteți ajuta renii să stabilească cine este cine, unde locuiește și ce cadou a cerut fiecare copil?



#### CLUES:

1. Nici Kliff, nici Jennifer nu locuiește în Kent.
2. Tanarul Crawford își dorește poneiul.
3. Tanarul Rowlands nu trăiește în Yorkshire și nu își dorește o bicicletă.
4. Cadoul care este pregătit pentru a fi dus la Cornwall este un calculator.
5. Calculatorul nu este pentru copilul cu numele Rowlands.
6. Liz, care își dorește setul de pictură, nu este din Kent. Numele ei nu este Jamison.
7. Copilul care trăiește în Essex își dorește o chitară.
8. Alan trăiește în Columbia.
9. Numele lui Sarah nu este Crawford.
10. Numele lui Jennifer este Feather.



### 3.2 Tabel de adevar

First	Last	City	Gift
Alan	Crawford	Cumbria	Pony
Cliff	Rowlands	Essex	Guitar
Jennifer	Feather	Cornwall	Computer
Liz	Northey	Yorkshire	Paint Set
Sarah	Jamison	Kent	Bicycle

### 3.3 Cod

#### 3.3.1 Modelare KB

```
¬Kent(Cliff).
¬Kent(Jennifer).
lastCrawford(x) → pony(x).
lastRowlands(x) → ¬Yorkshire(x).
lastRowlands(x) → ¬bicycle(x).
Cornwall(x) → computer(x).
computer(x) → ¬lastRowlands(x).
paintSet(Liz).
¬Kent(Liz).
¬lastJamison(Liz).
Essex(x) → guitar(x).
Cumbria(Alan).
¬lastCrawford(Sarah).
lastFeather(Jennifer).
```

#### 3.3.2 Implementare si testare incrementală

Implementarea si testarea incrementală s-a realizat prin testarea treptată in Prover9 a unor tinte(goal-uri) in ordinea in care ele s-au putut deduce.

```
lastCrawford(a).
```

```
Cumbria(a) & pony(a) & lastCrawford(a).
lastRowlands(b) & Essex(b) & guitar(b).
lastFeather(c) & Cornwall(c) & computer(c).
lastNorthey(d) & Yorkshire(d) & paintSet(d).
lastJamison(e) & Kent(e) & bicycle(e).
```

In primul rand, dupa ce am dedus din KB unele indicii si relatii, plecam de la premiza ca numele lui Allan este Rowlands. Conform clue-ului numarul 3, puteam face 4 alegeri, am ales s-o facem pe aceasta. Testam

in prover9 pentru a garanta rezolvarea acestui puzzle plecand de la premisa respectiva. Prover9 ne confirma corectitudinea premizei si de aici incolo deciziile se iau pe baza de rationamet logic. La final, dupa parcurgere, vom testa daca ce am dedus este corect ca ansamblu.

### **3.4 Pareri personale**

Prin acest tip de puzzle s-a exersat lucrul in FOL cu ceea ce numim constante. Aici avem o multitudine de relatii care contin constante, nu variabile, cum eram obisnuiti, exemplu: `-Kent(Cliff)`, `-Kent(Jennifer)`, `lastFeather(Jennifer)`.



## 4 Climber puzzle

### 4.1 Descriere puzzle

Patru exploratori au urcat munți diferiți în ani diferiți. Fiecare a petrecut câteva zile pentru a obține un astfel de triumf. Aflați care munte a fost explorat de Nicholas și în ce an.



#### CLUES:

1. Exploratorul care are nevoie de 10 zile să escaladeze un munte, nu a fost exploratorul care a urcat pe Everest în 1970.
2. Expeditia din 1970 a durat mai mult decât cea din 1990.
3. Dintre exploratorul Otto și exploratorul Ludwig, unul a plecat în expediție în anul 1990 și altul a escaladat Kilimanjaro.
4. Aconcagua a avut nevoie de 5 zile în plus pentru expediție decât expediția care a avut loc în anul 1970.
5. Expeditia de pe munte din anul 1980 a avut nevoie de 5 zile în plus decât expediția realizată de către Otto.
6. Everest a fost escaladat de Edmund sau în anul 1980.
7. Expeditia de pe munte din anul 1990 a avut nevoie cu 5 zile mai puțin decât expediția realizată de Edmund.

### 4.2 Tabel de adevăr

Mountain	Days	Year	Explorer
Aconcagua	20	1980	Nicholas
Everest	10	2000	Edmund
Kilimanjaro	15	1970	Otto
Vinson	5	1990	Ludwig

## 4.3 Cod

### 4.3.1 Modelare KB

```
Acongua(a).
Everest(b).
Kilimanjaro(c).
Vinson(d).

days10(x) -> (Everest(x) & -year1970(x)) | (-Everest(x) & year1970(x)).

year1970(x) & year1990(y) -> (days20(x) & days15(y)) |
                               (days20(x) & days10(y)) |
                               (days20(x) & days5(y)) |
                               (days15(x) & days10(y)) |
                               (days15(x) & days5(y)) |
                               (days10(x) & days5(y)).

Ludwig(x) & Otto(y) -> (year1990(x) & Kilimanjaro(y)) |
                      (year1990(y) & Kilimanjaro(x)).

Acongua(x) & year1970(y) -> (days10(x) & days5(y)) |
                             (days15(x) & days10(y)) |
                             (days20(x) & days15(y)).

year1980(x) & Otto(y) -> (days10(x) & days5(y)) |
                         (days15(x) & days10(y)) |
                         (days20(x) & days15(y)).

Everest(x) -> (Edmund(x) & -year1980(x)) | (-Edmund(x) & year1980(x)).

Edmund(x) & year1990(y) -> (days10(x) & days5(y)) |
                             (days15(x) & days10(y)) |
                             (days20(x) & days15(y)).
```

### 4.3.2 Implementare si testare incrementală

Implementarea si testarea incrementală s-a realizat prin testarea treptată în Prover9 a unor tinte(goal-uri) în ordinea în care ele s-au putut deduce.

```
days20(a) & year1980(a) & Nicholas(a).
days10(b) & year2000(b) & Edmund(b).
days15(c) & year1970(c) & Otto(c).
days5(d) & year1990(d) & Ludwig(d).
```

## 4.4 Pareri personale

Acest puzzle impune o abordare destul de eficientă în ceea ce privește modelarea relațiilor și implementarea lor. La o primă vedere, clue-ul 7, spre exemplu, ar avea nevoie de o funcție pentru a putea fi implementat: lessThan5days, ceea ce ar fi destul de dificil atât de implementat, cât și de mapat. Astfel, folosindu-ne de operațiile de sau logic (!) am exprimat o serie de relații compatibile cererii celor două: expediția din 1990 și expediția realizată de Edmund.

## 5 Pasta and Wine puzzle

### 5.1 Descriere puzzle

Cinci prieteni planifică o cină împreună. Fiecare se bucură de diferite tipuri de paste și vinuri din diferite țări. Aflați ce paste îi plac cel mai mult lui Holly.



#### CLUES:

1. Femeia care este în tricou alb este exact lângă femeia care apreciază vinurile din Lombardia.
2. Doamna Miller este undeva între doamna Davis și doamna Brown, în această ordine.
3. Cea mai tânără femeie se afla pe poziția 3.
4. Femeia de 45 de ani este undeva în dreapta femeii care poartă tricou roșu.
5. Femeia careia îi plac vinurile din Chile, îi plac și pastecele farfalle.
6. Pe prima poziție se afla femeia careia îi plac vinurile argentine.
7. Andrea este exact la dreapta femeii de 35 de ani.
8. Femeia care poartă tricou albastru este undeva între doamna Davis și Holly, în această ordine.
9. Victoria se afla lângă Leslie.
10. Femeia care poartă tricou roșu este undeva la stanga femeii careia îi plac vinurile australiene.
11. Doamna Wilson este exact lângă femeia de 30 ani.
12. Leslie este exact la stanga femeii de 30 ani.
13. Holly este undeva la dreapta femeii care poartă tricou roșu.
14. Doamna Brown este exact la stanga lui July.
15. Celei mai tinere femei îi plac pennele.
16. Doamna Wilson poartă tricou alb.
17. Femeia careia îi place Lasagnea este undeva între femeia careia îi plac vinurile italiene și femeia careia îi plac spaghetti, în această ordine.
18. Pe poziția doi se afla femeia care poartă tricou albastru.
19. Femeii de 40 ani îi place lasagnea,
20. Doamna Lopes se afla pe poziția 5.
21. Femeia careia îi plac vinurile australiene este undeva între Victoria și femeia careia îi plac vinurile din Franța, în această ordine.
22. Femeia care poartă tricou galben este exact în stanga femeii de 35 ani.

## 5.2 Tabel de adevar

yellow	blue	red
Victoria	Leslie	Andrea
Davis	Miller	Brown
ravioli	farfalle	penne
Argentine	Chilean	Italian
50 years	35 years	30 years

  

white	green
Julie	Holly
Wilson	Lopes
lasagne	spaghetti
Australian	French
40 years	45 years

## 5.3 Cod - Implementare I

### 5.3.1 Modelare KB

```

shirtWhite(x) & wineItalian(y) -> neighbor(x, y).
Miller(x) & Davis(y) & Brown(z) -> right(y, x) & right(x, z).
Age30(c).
Age45(x) & shirtRed(y) -> right(y, x).
wineChilean(x) -> pastaFarfalle(x).
wineArgentine(a).
Andrea(x) & Age35(y) -> rightneighbor(y, x).
shirtBlue(x) & Davis(y) & Holly(z) -> right(y, x) & right(x, z).
Victoria(x) & Leslie(y) -> neighbor(x, y).
shirtRed(x) & wineAustralian(y) -> right(x, y).
Wilson(x) & Age30(y) -> neighbor(x, y).
Leslie(x) & Age30(y) -> rightneighbor(x, y).
Holly(x) & shirtRed(y) -> right(y, x).
Brown(x) & Julie(y) -> rightneighbor(x, y).
Age30(x) -> pastaPene(x).
Wilson(x) -> shirtWhite(x).
pastaLasagne(x) & wineItalian(y) & pastaSpagetti(z) -> right(y, x) & right(x, z).

shirtBlue(b).
Age40(x) -> pastaLasagne(x).
Lopes(e).

```

```
wineAustralian(x) & Victoria(y) & wineFrench(z) -> right(y, x) & right(x, z).

shirtYellow(x) & Age35(y) -> rightneighbor(x, y).
```

### 5.3.2 Implementare si testare incrementală

Implementarea si testarea incrementală s-a realizat prin testarea treptată în Prover9 a unor tinte(goal-uri) în ordinea în care ele s-au putut deduce.

```
Leslie(b).
Victoria(a).
Age40(d).
pastaSpagetti(e).
Lopes(e).
Farfalle(b).
```

Initial, deducem logic unele aspecte din KB -ul nostru. Din faptul ca femeia de 30 de ani se afla pe pozitia 3, iar Leslie se afla exact la stanga acesteia, deducem ca Leslie se afla pe pozitia 2(b). Prover9 certifica acest lucru. Mergând în continuare cu rationamentul logic, va trebui sa alegem o premiza: Victoria se afla pe prima pozitie. Victoria fiind exact langa Leslie, puteam alege ca Victoria sa fie pe pozitia 1 sau pe pozitia 3. Am ales ca Victoria sa fie pe pozitia 1 (a). Testam aceasta premiza cu prover9. El ne garanteaza corectitudinea aceste alegeri. Pe urma, toate celelalte se deduc prin rationament logic.

## 5.4 Cod - Implementare II

### 5.4.1 Modelare KB

```
neighbors(White, Italian).
somewhereBetween(Davis, Miller, Brown).
Age30 = 2.
somewhereRight(Age45, Red).
Chilean = farfalle.
Argentina = 0.
right_neighbor(Age35, Andrea).
somewhereBetween(Davis, Blue, Holly).
neighbors(Victoria, Leslie).
somewhereLeft(Red, Australian).
neighbors(Wilson, Age30).
right_neighbor(Leslie, Age30).
somewhereRight(Holly, Red).
right_neighbor(Brown, Julie).
Age30 = penne.
Wilson = White.
somewhereBetween(Italian, lasagne, spaghetti).
Blue = 1.
Age40 = lasagne.
Lopes = 4.
somewhereBetween(Victoria, Australian, French).
right_neighbor(Yellow, Age35).
```

## 5.5 Pareri personale

Puzzle-ul acesta pune în evidență lucrul cu functii. Astfel, am abordat două modalități de implementare, una specifică mace4 (implementarea II) și una specifică prover9 (implementarea I). După cum se poate observa, a

doua metoda este mult mai rapida din punct de vedere a cantitatii de cod, dar are un dezavantaj: aceasta nu poate fi testata cu prover9 si deci, nu poate fi testata incremental. Totodata, prezinta si un avantaj: usurinta de definire a functiilor datorita flagului : set (arithmetic), specific doar pe Linux. (ex. : somewhereRight(x, y) <-> x > y. )

## 6 Party puzzle

### 6.1 Descriere puzzle

Intr-un oras exista persoane cu trei functii: scriitori, actori si jurnalisti. Acestia pot fi invitati la o petrecere si pot ajunge la timp sau nu. Conform indicilor, aflati care este persoana care a ajuns la timp la aceasta petrecere!



#### CLUES:

1. In oras exista scriitori, jurnalisti si actori.
2. Orice actor e invitat la petrecere.
3. Orice jurnalist e invitat la petrecere .
4. Cativa scriitori sunt invitati la petrecere.
5. Orice invitat la petrecere care e actor si jurnalist va intarzia.
6. Cativa invitati ajung la petrecere la timp.
7. Cativa invitati nu sunt nici actori, nici jurnalisti.

#### QUESTION:

Persoana care ajunge la timp la petrecere este un scriitor?

### 6.2 Cod

#### 6.2.1 Modelare KB

```
city(x) -> actor(x) & journalist(x) & writer(x).
all x (actor(x) & invited(x)).
all x (journalist(x) & invited(x)).
exists x (writer(x) & invited(x)).
all x ( actor(x) & journalist(x) ) & invited(x) -> late(x).
exists x (¬ late(x)).
exists x (invited(x) & ¬actor(x) & ¬journalist(x)).
```

#### 6.2.2 Testare incrementală

Testarea s-a realizat in Prover9 prin verificarea unui singur goal: este persoana care nu intarzie un scriitor?. Prin testarea acestui goal raspundem la intrebarea atasata puzzle-ului: ce persoana ajunge la timp?

```
exists x ( writer (x) & invited(x) & ¬late(x)).
```

### 6.3 Pareri personale

Puzzle-ul prezentat este diferit de celelalte tipuri de puzzle-uri, intrucat impune o alta abordare logica. Putem spune ca acest puzzle deschide viziunea unei ghicitori, prin care conform unor axiome, afirmatii care se iau drept adevarate fara a se demonstra, deducem si raspundem la ghicitoare.