

Inteligența Artificială - Proiect 2

-Probleme Prover9 -

Băck Cosmin-Robert
Bălc Horia-Ovidiu

9/12/2020

1 Problema cavalerilor si a escrocilor

1.1 Cerintele problemei

Problema cavalerilor si a escrocilor are loc pe o insula cu doua tipuri de oameni: cavaleri, care spun mereu adevarul si escroci, care mint tot timpul.

Pe aceasta insula te intalnesti cu trei persoane: Jim, Jon si Joe.

Jim spune: "Cel putin una din urmatoarele propozitii este adevarata: Joe este un escroc sau eu sunt un cavaler."

Jon spune: "Jim ar spune ca eu sunt un escroc."

Joe spune: "Niciunul dintre Jim si Jon nu este cavaler."

Cine este cavaler si cine este escroc?

1.2 Analiza problemei

Cel mai simplu este sa incepem cu testarea propozitiei lui Joe.

Cazul 1: Daca Joe este cavaler, atunci Jim si Jon sunt escroci. In acest caz, propozitia lui Jim este falsa, ceea ce confirma faptul ca el este escroc. Propozitia lui Jon este si ea falsa, deoarece Jim, fiind escroc, si Jon asemenea, ar spune ca Jon este cavaler, ceea ce confirma faptul ca si Jon este escroc. In concluzie, Jim si Jon sunt escroci si Joe este cavaler.

Cazul 2: Daca Joe este escroc, inseamna ca cel putin unul dintre Jim si Jon este cavaler. Daca Jim este cavaler, atunci propozitia lui este adevarata, ceea ce confirma faptul ca el este cavaler. Insa, in acest caz, Jon nu poate sa fie nici cavaler si nici escroc, deoarece daca ar fi cavaler, atunci propozitia lui ar fi falsa si daca ar fi escroc, atunci propozitia lui ar fi adevarata, cazuri care duc la contradictie. Deci Jim nu poate sa fie cavaler, asadar Jon trebuie sa fie cavaler. Daca Jon este cavaler, atunci propozitia lui este adevarata, ceea ce il face pe Jim escroc. Insa Jim nu poate fi escroc, deoarece propozitia spusa de el este adevarata.

In concluzie, singura solutie este: Jim si Jon - escroci si Joe - cavaler.

1.3 Implementarea problemei

Cod:

```
1 knight(x) | knave(x).
2
3 knight(Jim) -> -knave(Jim).
4 knight(Joe) -> -knave(Joe).
5 knight(Jon) -> -knave(Jon).
6
7 knight(Jim) -> knave(Joe) | knight(Jim).
8 knave(Jim) -> -(knave(Joe) | knight(Jim)).
9
10 knight(Joe) -> knave(Jim) & knave(Jon).
11 knave(Joe) -> knight(Jim) | knight(Jon).
12
13 knight(Jon) & knight(Jim) -> knave(Jon).
14 knight(Jon) & knave(Jim) -> knight(Jon).
15 knave(Jon) & knight(Jim) -> knight(Jon).
16 knave(Jon) & knave(Jim) -> knave(Jon).
```

1.4 Interpretare rezultate cu Mace4:

```
interpretation( 2, [number = 1,seconds = 0], [
    function(Jim, [0]),
    function(Joe, [1]),
    function(Jon, [0]),
    relation(knave(_), [1,0]),
    relation(knight(_), [0,1]))).
```

Figure 1: Problema cavalierilor si a escrocilor - interpretare rezultate

Interpretare:

Jim si Jon sunt mapati la pozitia 0, iar Joe este mapat la pozitia 1.

Relatia knave(escroc) pentru pozitia 0 are valoarea 1, iar pentru pozitia 1 are valoarea 0, deci Jim si Jon sunt knaves, iar Joe nu este knave.

Relatia knight(cavaler) pentru pozitia 0 are valoarea 0, iar pentru pozitia 1 are valoarea 1, deci Jim si Jon nu sunt knights, iar Joe este knight.

2 Test auto-referential

2.1 Cerintele problemei

Un test auto-referential este un test cu intrebari cu varianta unica, raspunsurile depinzand unele de altele. Se cere solutia urmatorului test auto-referential:

1. Varianta care nu este un raspuns la nicio alta intrebare este
 - A. ☐ A
 - B. ☐ B
 - C. ☐ C
 - D. ☐ D
2. Singura intrebare cu acelasi raspuns ca si la aceasta este
 - A. ☐ 3
 - B. ☐ 1
 - C. ☐ 5
 - D. ☐ 4
3. Raspunsul la intrebarea numarul 5 este
 - A. ☐ B
 - B. ☐ D
 - C. ☐ A
 - D. ☐ C
4. Prima intrebare a carei raspuns este A este
 - A. ☐ 2
 - B. ☐ 3
 - C. ☐ 4
 - D. ☐ 5
5. Raspunsul la intrebarea numarul 3 este
 - A. ☐ C
 - B. ☐ B
 - C. ☐ D
 - D. ☐ A

Figure 2: Test Auto-Referential

2.2 Analiza problemei

Pentru a rezolva acest test, vom incepe cu intrebarile 3 si 5. Solutiile posibile sunt: D, respectiv C si C, respectiv A.

Cazul 1: 3.D 5.C - in acest caz, continuam cu intrebarea 4, a carei raspuns trebuie sa fie A. In acest caz, raspunsul la intrebarea 2 trebuie sa fie A, ceea ce ar insemna ca raspunsul la intrebarea 3 este tot A (contradictie).

Cazul 2: 3.C 5.A - in acest caz, continuam cu intrebarea 4, a carei raspuns trebuie sa fie A sau D. Daca raspunsul la intrebarea 4 este A, atunci raspunsul la intrebarea 2 trebuie sa fie A, ceea ce ar insemna ca raspunsul la intrebarea 3 este tot A (contradictie). Daca raspunsul la intrebarea 4 este D, atunci raspunsul la intrebarea 2 poate sa fie B sau D. Daca raspunsul la intrebarea 2 este B, atunci raspunsul la intrebarea 1 este B (contradictie). Daca raspunsul la intrebarea 2 este D, atunci raspunsul la intrebarea 1 este B.

In concluzie, solutia testului este:

1. Varianta care nu este un raspuns la nicio alta intrebare este
 - A. ✗ A
 - B. ✓ B
 - C. ✗ E
 - D. ✗ D
2. Singura intrebare cu acelasi raspuns ca si la aceasta este
 - A. ✗ 3
 - B. ✗ 1
 - C. ✗ 5
 - D. ✓ 4
3. Raspunsul la intrebarea numarul 5 este
 - A. ✗ B
 - B. ✗ D
 - C. ✓ A
 - D. ✗ E
4. Prima intrebare a carei raspuns este A este
 - A. ✗ 2
 - B. ✗ 3
 - C. ✗ 4
 - D. ✓ 5
5. Raspunsul la intrebarea numarul 3 este
 - A. ✓ C
 - B. ✗ B
 - C. ✗ D
 - D. ✗ A

Figure 3: Test Auto-Referential - Solutia

2.3 Implementarea problemei

Cod:

```
1  a(x) <-> -b(x) & -c(x) & -d(x).
2  b(x) <-> -a(x) & -c(x) & -d(x).
3  c(x) <-> -a(x) & -b(x) & -d(x).
4  d(x) <-> -a(x) & -b(x) & -c(x).
5
6  a(q1) <-> -a(q2) & -a(q3) & -a(q4) & -a(q5).
7  b(q1) <-> -b(q2) & -b(q3) & -b(q4) & -b(q5).
8  c(q1) <-> -c(q2) & -c(q3) & -c(q4) & -c(q5).
9  d(q1) <-> -d(q2) & -d(q3) & -d(q4) & -d(q5).
10
11 a(q2) -> a(q3) & -a(q1) & -a(q4) & -a(q5).
12 b(q2) -> b(q1) & -b(q3) & -b(q4) & -b(q5).
13 c(q2) -> c(q5) & -c(q1) & -c(q3) & -c(q4).
14 d(q2) -> d(q4) & -d(q1) & -d(q3) & -d(q5).
15
16 a(q3) <-> b(q5).
17 b(q3) <-> d(q5).
18 c(q3) <-> a(q5).
19 d(q3) <-> c(q5).
20
21 a(q4) <-> a(q2) & -a(q1).
22 b(q4) <-> a(q3) & -a(q2) & -a(q1).
23 c(q4) <-> a(q4) & -a(q3) & -a(q2) & -a(q1).
24 d(q4) <-> a(q5) & -a(q4) & -a(q3) & -a(q2) & -a(q1).
25
26 a(q5) <-> c(q3).
27 b(q5) <-> b(q3).
28 c(q5) <-> d(q3).
29 d(q5) <-> a(q3).
```

2.4 Interpretare rezultate cu Mace4:

```
interpretation( 4, [number = 1,seconds = 0], [
    function(q1, [0]),
    function(q2, [1]),
    function(q3, [2]),
    function(q4, [1]),
    function(q5, [3]),
    relation(a(_), [0,0,0,1]),
    relation(b(_), [1,0,0,0]),
    relation(c(_), [0,0,1,0]),
    relation(d(_), [0,1,0,0])]).
```

Figure 4: Test auto-referential - interpretare rezultate

Interpretare:

Intrebarea 1 este mapata la valoarea 0, intrebarile 2 si 4 sunt mapate la valoarea 1, intrebarea 3 este mapata la valoarea 2 si intrebarea 5 este mapata la valoarea 3.

Relatia a, care corespunde raspunsului A este 1 doar pentru valoarea 3, adica pentru intrebarea 5.

Relatia b, care corespunde raspunsului B este 1 doar pentru valoarea 0, adica pentru intrebarea 1.

Relatia c, care corespunde raspunsului C este 1 doar pentru valoarea 2, adica pentru intrebarea 3.

Relatia d, care corespunde raspunsului D este 1 doar pentru valoarea 1, adica pentru intrebarile 2 si 4.

3 Problema usilor

3.1 Cerintele problemei

Te afli într-o camera cu trei usi. În spatele unei singure usi se afla libertatea. Fiecare usa are câte o propoziție scrisă pe ea. Cel puțin o propoziție este adevărată și cel puțin una este falsă.

Prima usa: "Libertatea este în spatele acestei usi."

A doua usa: "Libertatea nu este în spatele acestei usi."

A treia usa: "Libertatea nu este în spatele celei de-a doua usi."

Care usa duce la libertate?

3.2 Analiza problemei

Cazul 1: Libertatea este în spatele primei usi.

În acest caz, prima propoziție este adevărată. Propozițiile 2 și 3 sunt și ele adevărate, deoarece libertatea nu este în spatele celei de-a doua usi. Toate cele trei propoziții sunt adevărate, ceea ce duce la contradicție cu cerința (trebuie cel puțin o propoziție să fie falsă).

Cazul 2: Libertatea este în spatele celei de-a doua usi.

În acest caz, prima propoziție este falsă deoarece libertatea nu este în spatele primei usi. Propozițiile 2 și 3 sunt și ele false, deoarece libertatea se afla în spatele celei de-a doua usi. Toate cele trei propoziții sunt false, ceea ce duce la contradicție cu cerința (trebuie cel puțin o propoziție să fie adevărată).

Cazul 3: Libertatea este în spatele celei de-a treia usi.

În acest caz, prima propoziție este falsă deoarece libertatea nu este în spatele primei usi. Propozițiile 2 și 3 sunt adevărate, deoarece libertatea nu este în spatele celei de-a doua usi.

În concluzie, libertatea se afla în spatele celei de-a treia usi.

3.3 Implementarea problemei

Cod:

```
1  libertate1 | libertate2 | libertate3.
2  adevar1 | adevar2 | adevar3 | (adevar1 & adevar2) | (adevar3 & adevar2) | (adevar3 & adevar1)
3
4  libertate1 -> -libertate2.
5  libertate1 -> -libertate3.
6  libertate2 -> -libertate1.
7  libertate2 -> -libertate3.
8  libertate3 -> -libertate1.
9  libertate3 -> -libertate2.
10
11 adevar1 <-> libertate1.
12 adevar2 <-> -libertate2.
13 adevar3 <-> -libertate2.
14
15 -adevar1 <-> -libertate1.
16 -adevar2 <-> libertate2.
17 -adevar3 <-> libertate2.
18
19 adevar1 -> -adevar2 | -adevar3.
20 adevar2 -> -adevar1 | -adevar3.
21 adevar3 -> -adevar1 | -adevar2.
```

3.4 Interpretare rezultate cu Mace4:

```
interpretation( 2, [number = 1,seconds = 0], [
    relation(adevar1, [0]),
    relation(adevar2, [1]),
    relation(adevar3, [1]),
    relation(libertate1, [0]),
    relation(libertate2, [0]),
    relation(libertate3, [1])]).
```

Figure 5: Problema usilor - interpretare rezultate

Interpretare:

Relatia adevar1 este 0, adica prima propozitie este falsa.

Relatiile adevar2 si adevar3 sunt 1, adica propozitiile 2 si 3 sunt adevarate.

Relatia libertate1 si libertate2 sunt 0, iar relatia libertate3 este 1, adica libertatea se gaseste in spatele celei de-a treia usi.

4 Problema spionilor

4.1 Cerintele problemei

Problema spionilor se refera la o echipa de 5 spioni, fiecare spion avand o cravata de o anumita culoare, un nume, o tara in care urmeaza sa mearga, un accesoriu, o aptitudine si o varsta. Toate aceste atribute sunt unice pentru fiecare spion. Spionii sunt asezati intr-un sir.

Cravatele sunt de 5 culori: negru, albastru, verde, mov si rosu.

Numele lor sunt: Austin, James, Jason, Stan si Sterling.

Tarile in care vor merge sunt: Australia, Brazilia, Germania, Libia si Rusia.

Accesoriile lor sunt: ceas, stilou, telefon, inel si umbrela.

Aptitudinile lor sunt: computer hacking, deghizare, condus, arte martiale si parkour.

Varstele lor sunt: 25 de ani, 30 de ani, 35 de ani, 40 de ani si 45 de ani.

Se dau urmatoarele informatii:

Austin e langa spionul cu cravata neagra.

Maestrul in deghizari este chiar in dreapta spionului care are umbrela.

Spionul de 35 de ani se duce in Libia.

James are 25 de ani.

Spionul care se duce in Australia este langa spionul care are ca aptitudine parkour-ul.

James este in chiar in dreapta spionului care are ceas.

Spionul care are umbrela este undeva intre spionul de 40 de ani si Austin, in aceasta ordine.

Stan este langa agentul care se duce in Rusia.

Sterling este la unul dintre capete.

Barbatul care poarta cravata rosie are 40 de ani.

Spionul care se duce in Brazilia este chiar in stanga spionului de 40 de ani.

Jason este chiar in stanga lui Austin.

Experul in condus este langa spionul de 30 de ani.

Spionul de 35 de ani este langa spionul care se duce in Australia.

Spionul specializat in computer hacking este chiar in stanga spionului de 35 de ani.

Spionul care poarta cravata mov este langa spionul specializat in computer hacking.

Austin are 30 de ani.

Spionul care are telefon este chiar in stanga spionului care se duce in Libia.

Spionul care are inel este undeva la dreapta spionului care poarta cravata mov.

In pozitia a doua este spionul care poarta cravata verde.

Spionul care merge in Australia este chiar in dreapta spionului de 30 de ani.

Care spion este specializat in arte martiale?

4.2 Analiza problemei

Incepem prin a pune cravata verde pe pozitia a doua.

Verificam pozitiile lui Austin: stim ca trebuie ca unul dintre spioni sa fie intre spionul de 40 de ani si Austin, deci Austin se afla pe pozitiile 3, 4 sau 5. Austin are 30 de ani si stim ca spionul care merge in Australia este in dreapta sa, deci Austin se afla pe pozitiile 3 sau 4. Spionul de 35 de ani este langa spionul care merge in Australia, ceea ce inseamna ca spionul care merge in Australia este pe pozitia 4, iar Austin este pe pozitia 3. Spionul de 35 de ani este pe pozitia 5 si merge in Libia. Jason este in stanga lui Austin, deci pe pozitia 2.

Austin sta langa spionul cu cravata neagra, deci pe pozitia 4 se afla cravata neagra.

Spionul care are telefon este in stanga celui care merge in Libia, deci acesta se afla pe pozitia 4.

Spionul care e specializat in computer hacking este in stanga celui de 35 de ani, deci acesta se afla tot pe pozitia 4.

Austin se afla langa spionul specializat in condus, deci cel din urma este pe pozitia 2.

Spionul care are umbrela este intre cel de 40 de ani si Austin, deci cel cu umbrela este pe pozitia 2 si cel de 40 de ani este pe pozitia 1.

Cel de 40 de ani are cravata rosie, deci cravata rosie este pe pozitia 1.

Spionul cu cravata mov este langa cel specializat in computer hacking, deci pe pozitiile 3 sau 5. Spionul care are inel este undeva in dreapta celui cu cravata mov, deci cel cu cravata mov trebuie sa fie pe pozitia a 3-a, iar cel cu inel pe pozitia 5.

Prin eliminare, spionul cu cravata albastra este pe pozitia a 5-a.

James este in dreapta spionului care are ceas, deci James este pe pozitia 4, iar cel care are ceas este pe pozitia 3.

James are 25 de ani. Prin eliminare, cel care are 45 de ani este pe pozitia a 2-a.

Prin eliminare, spionul care are stilou este pe pozitia 1.

Cel care merge in Brazilia este in stanga celui de 45 de ani, deci pe pozitia 1.

Spionul specializat in deghizare este in dreapta celui cu umbrela, deci pe pozitia a 3-a.

Spionul care se duce in Australia este langa cel specializat in parkour, deci cel din urma este pe pozitia a 5-a.

Stan este langa spionul care merge in Rusia, deci Stan este pe pozitia 1, iar cel care merge in Rusia este pe pozitia a 2-a.

Sterling este la unul dintre capete, deci pe pozitia 5.

Prin eliminare, cel care merge in Germania este pe pozitia a 3-a, iar cel care e specializat in arte martiale este pe pozitia 1.

	Pozitia 1	Pozitia 2	Pozitia 3	Pozitia 4	Pozitia 5
Cravata	rosie	verde	mov	neagra	albastra
Nume	Stan	Jason	Austin	James	Sterling
Tara	Brazilia	Rusia	Germania	Australia	Libia
Accesorii	stilou	umbrela	ceas	telefon	inel
Aptitudine	arte martiale	condus	deghizare	computer hacking	parkour
Varsta	40 de ani	45 de ani	30 de ani	25 de ani	35 de ani

Figure 6: Problema spionilor

In concluzie, cel care practica arte martiale se afla pe pozitia 1.

4.3 Implementarea problemei

Cod:

```

1  differentFrom(a,b).
2  differentFrom(a,c).
3  differentFrom(a,d).
4  differentFrom(a,e).
5  differentFrom(b,c).
6  differentFrom(b,d).
7  differentFrom(b,e).
8  differentFrom(c,d).
9  differentFrom(c,e).
10 differentFrom(d,e).
11 differentFrom(x,y) -> differentFrom(y,x).
12
13 rightneighbor(a,b).
14 rightneighbor(b,c).
15 rightneighbor(c,d).
16 rightneighbor(d,e).
17 -rightneighbor(a,a).
18 -rightneighbor(a,c).
19 -rightneighbor(a,d).
20 -rightneighbor(a,e).
21 -rightneighbor(b,a).
22 -rightneighbor(b,b).
23 -rightneighbor(b,d).
24 -rightneighbor(b,e).
25 -rightneighbor(c,a).
26 -rightneighbor(c,b).
27 -rightneighbor(c,c).
28 -rightneighbor(c,e).
29 -rightneighbor(d,a).
```

```

30 -rightneighbor(d,b).
31 -rightneighbor(d,c).
32 -rightneighbor(d,d).
33 -rightneighbor(e,a).
34 -rightneighbor(e,b).
35 -rightneighbor(e,c).
36 -rightneighbor(e,d).
37 -rightneighbor(e,e).
38 rightneighbor(x,y) | rightneighbor(y,x) <-> neighbor(x,y).
39
40 somewhereRight(a,b).
41 somewhereRight(a,c).
42 somewhereRight(a,d).
43 somewhereRight(a,e).
44 somewhereRight(b,c).
45 somewhereRight(b,d).
46 somewhereRight(b,e).
47 somewhereRight(c,d).
48 somewhereRight(c,e).
49 somewhereRight(d,e).
50 -somewhereRight(a,a).
51 -somewhereRight(b,a).
52 -somewhereRight(b,b).
53 -somewhereRight(c,a).
54 -somewhereRight(c,b).
55 -somewhereRight(c,c).
56 -somewhereRight(d,a).
57 -somewhereRight(d,b).
58 -somewhereRight(d,c).
59 -somewhereRight(d,d).
60 -somewhereRight(e,a).
61 -somewhereRight(e,b).
62 -somewhereRight(e,c).
63 -somewhereRight(e,d).
64 -somewhereRight(e,e).
65
66 between(b,a,c).
67 between(b,a,d).
68 between(b,a,e).
69 between(c,a,d).
70 between(c,a,e).
71 between(c,b,d).
72 between(c,b,e).
73 between(d,a,e).
74 between(d,b,e).
75 between(d,c,e).

```

```

76  -between(a,a,a).
77  -between(a,a,b).
78  -between(a,a,c).
79  -between(a,a,d).
80  -between(a,a,e).
81  -between(a,b,a).
82  -between(a,b,b).
83  -between(a,b,c).
84  -between(a,b,d).
85  -between(a,b,e).
86  -between(a,c,a).
87  -between(a,c,b).
88  -between(a,c,c).
89  -between(a,c,d).
90  -between(a,c,e).
91  -between(a,d,a).
92  -between(a,d,b).
93  -between(a,d,c).
94  -between(a,d,d).
95  -between(a,d,e).
96  -between(a,e,a).
97  -between(a,e,b).
98  -between(a,e,c).
99  -between(a,e,d).
100 -between(a,e,e).
101 -between(b,a,a).
102 -between(b,a,b).
103 -between(b,b,a).
104 -between(b,b,b).
105 -between(b,b,c).
106 -between(b,b,d).
107 -between(b,b,e).
108 -between(b,c,a).
109 -between(b,c,b).
110 -between(b,c,c).
111 -between(b,c,d).
112 -between(b,c,e).
113 -between(b,d,a).
114 -between(b,d,b).
115 -between(b,d,c).
116 -between(b,d,d).
117 -between(b,d,e).
118 -between(b,e,a).
119 -between(b,e,b).
120 -between(b,e,c).
121 -between(b,e,d).

```

```

122 -between(b,e,e).
123 -between(c,a,a).
124 -between(c,a,b).
125 -between(c,a,c).
126 -between(c,b,a).
127 -between(c,b,b).
128 -between(c,b,c).
129 -between(c,c,a).
130 -between(c,c,b).
131 -between(c,c,c).
132 -between(c,c,d).
133 -between(c,c,e).
134 -between(c,d,a).
135 -between(c,d,b).
136 -between(c,d,c).
137 -between(c,d,d).
138 -between(c,d,e).
139 -between(c,e,a).
140 -between(c,e,b).
141 -between(c,e,c).
142 -between(c,e,d).
143 -between(c,e,e).
144 -between(d,a,a).
145 -between(d,a,b).
146 -between(d,a,c).
147 -between(d,a,d).
148 -between(d,b,a).
149 -between(d,b,b).
150 -between(d,b,c).
151 -between(d,b,d).
152 -between(d,c,a).
153 -between(d,c,b).
154 -between(d,c,c).
155 -between(d,c,d).
156 -between(d,d,a).
157 -between(d,d,b).
158 -between(d,d,c).
159 -between(d,d,d).
160 -between(d,d,e).
161 -between(d,e,a).
162 -between(d,e,b).
163 -between(d,e,c).
164 -between(d,e,d).
165 -between(d,e,e).
166 -between(e,a,a).
167 -between(e,a,b).

```

```

168 -between(e,a,c).
169 -between(e,a,d).
170 -between(e,a,e).
171 -between(e,b,a).
172 -between(e,b,b).
173 -between(e,b,c).
174 -between(e,b,d).
175 -between(e,b,e).
176 -between(e,c,a).
177 -between(e,c,b).
178 -between(e,c,c).
179 -between(e,c,d).
180 -between(e,c,e).
181 -between(e,d,a).
182 -between(e,d,b).
183 -between(e,d,c).
184 -between(e,d,d).
185 -between(e,d,e).
186 -between(e,e,a).
187 -between(e,e,b).
188 -between(e,e,c).
189 -between(e,e,d).
190 -between(e,e,e).
191
192
193 black(x) | blue(x) | green(x) | purple(x) | red(x).
194 austin(x) | james(x) | jason(x) | stan(x) | sterling(x).
195 australia(x) | brazil(x) | germany(x) | libya(x) | russia(x).
196 clock(x) | pen(x) | phone(x) | ring(x) | umbrella(x).
197 computerhacking(x) | disguise(x) | driving(x) | martialarts(x) | parkour(x).
198 age25(x) | age30(x) | age35(x) | age40(x) | age45(x).
199
200 black(x) & black(y) -> -differentFrom(x,y).
201 blue(x) & blue(y) -> -differentFrom(x,y).
202 green(x) & green(y) -> -differentFrom(x,y).
203 purple(x) & purple(y) -> -differentFrom(x,y).
204 red(x) & red(y) -> -differentFrom(x,y).
205
206 austin(x) & austin(y) -> -differentFrom(x,y).
207 james(x) & james(y) -> -differentFrom(x,y).
208 jason(x) & jason(y) -> -differentFrom(x,y).
209 stan(x) & stan(y) -> -differentFrom(x,y).
210 sterling(x) & sterling(y) -> -differentFrom(x,y).
211
212 australia(x) & australia(y) -> -differentFrom(x,y).
213 brazil(x) & brazil(y) -> -differentFrom(x,y).

```



```

214 germany(x) & germany(y) -> -differentFrom(x,y).
215 libya(x) & libya(y) -> -differentFrom(x,y).
216 russia(x) & russia(y) -> -differentFrom(x,y).
217
218 clock(x) & clock(y) -> -differentFrom(x,y).
219 pen(x) & pen(y) -> -differentFrom(x,y).
220 phone(x) & phone(y) -> -differentFrom(x,y).
221 ring(x) & ring(y) -> -differentFrom(x,y).
222 umbrella(x) & umbrella(y) -> -differentFrom(x,y).
223
224 computerhacking(x) & computerhacking(y) -> -differentFrom(x,y).
225 disguise(x) & disguise(y) -> -differentFrom(x,y).
226 driving(x) & driving(y) -> -differentFrom(x,y).
227 martialarts(x) & martialarts(y) -> -differentFrom(x,y).
228 parkour(x) & parkour(y) -> -differentFrom(x,y).
229
230 age25(x) & age25(y) -> -differentFrom(x,y).
231 age30(x) & age30(y) -> -differentFrom(x,y).
232 age35(x) & age35(y) -> -differentFrom(x,y).
233 age40(x) & age40(y) -> -differentFrom(x,y).
234 age45(x) & age45(y) -> -differentFrom(x,y).
235
236 austin(x) & black(y) -> neighbor(x,y).
237 disguise(x) & umbrella(y) -> rightneighbor(y,x).
238 age35(x) <-> libya(x).
239 james(x) <-> age25(x).
240 australia(x) & parkour(y) -> neighbor(x,y).
241 james(x) & clock(y) -> rightneighbor(y,x).
242 umbrella(x) & age40(y) & austin(z) -> between (x,y,z).
243 stan(x) & rusia(y) -> neighbor(x,y).
244 sterling(a) | sterling(e).
245 red(x) <-> age40(x).
246 brazil(x) & age45(y) -> rightneighbor(x,y).
247 jason(x) & austin(y) -> rightneighbor(x,y).
248 driving(x) & age30(y) -> neighbor(x,y).
249 age35(x) & australia(y) -> neighbor(x,y).
250 computerhacking(x) & age35(y) -> rightneighbor(x,y).
251 purple(x) & computerhacking(y) -> neighbor(x,y).
252 austin(x) <-> age30(x).
253 phone(x) & libya(y) -> rightneighbor(x,y).
254 ring(x) & purple(y) -> somewhereRight(y,x).
255 green(b).
256 australia(x) & age30(y) -> rightneighbor(y,x).

```

4.4 Interpretare rezultate cu Mace4:

```
interpretation( 5, [number = 1,seconds = 0], [
  function(a, [0]),
  function(b, [1]),
  function(c, [2]),
  function(d, [3]),
  function(e, [4]),
  relation(age25(_), [0,0,0,1,0]),
  relation(age30(_), [0,0,1,0,0]),
  relation(age35(_), [0,0,0,0,1]),
  relation(age40(_), [1,0,0,0,0]),
  relation(age45(_), [0,1,0,0,0]),
  relation(austin(_), [0,0,1,0,0]),
  relation(australia(_), [0,0,0,1,0]),
  relation(black(_), [0,0,0,1,0]),
  relation(blue(_), [0,0,0,0,1]),
  relation(brazil(_), [1,0,0,0,0]),
  relation(clock(_), [0,0,1,0,0]),
  relation(computerhacking(_), [0,0,0,1,0]),
  relation(disguise(_), [0,0,1,0,0]),
  relation(driving(_), [0,1,0,0,0]),
  relation(germany(_), [0,0,1,0,0]),
  relation(green(_), [0,1,0,0,0]),
  relation(james(_), [0,0,0,1,0]),
  relation(jason(_), [0,1,0,0,0]),
  relation(libya(_), [0,0,0,0,1]),
  relation(martialarts(_), [1,0,0,0,0]),
  relation(parkour(_), [0,0,0,0,1]),
  relation(pen(_), [1,0,0,0,0]),
  relation(phone(_), [0,0,0,1,0]),
  relation(purple(_), [0,0,1,0,0]),
  relation(red(_), [1,0,0,0,0]),
  relation(ring(_), [0,0,0,0,1]),
  relation(russia(_), [0,1,0,0,0]),
  relation(stan(_), [1,0,0,0,0]),
  relation(sterling(_), [0,0,0,0,1]),
  relation(umbrella(_), [0,1,0,0,0]),
```

Figure 7: Problema spionilor - interpretare rezultate

Interpretare:

Pozitia 1 este mapata la valoarea 0, pozitia 2 este mapata la valoarea 1, pozitia 3 este mapata la valoarea 2, pozitia 4 este mapata la valoarea 3, iar pozitia 5 este mapata la valoarea 4.

Relatia age25 este 1 doar pentru valoarea 3, adica pentru pozitia 4.
Relatia age30 este 1 doar pentru valoarea 2, adica pentru pozitia 3.
Relatia age35 este 1 doar pentru valoarea 4, adica pentru pozitia 5.
Relatia age40 este 1 doar pentru valoarea 0, adica pentru pozitia 1.
Relatia age45 este 1 doar pentru valoarea 1, adica pentru pozitia 2.
Relatia austin este 1 doar pentru valoarea 3, adica pentru pozitia 4.
Relatia australia este 1 doar pentru valoarea 3, adica pentru pozitia 4.
Relatia black este 1 doar pentru valoarea 3, adica pentru pozitia 4.
Relatia blue este 1 doar pentru valoarea 4, adica pentru pozitia 5.
Relatia brazil este 1 doar pentru valoarea 0, adica pentru pozitia 1.
Relatia clock este 1 doar pentru valoarea 2, adica pentru pozitia 3.
Relatia computerhacking este 1 doar pentru valoarea 3, adica pentru pozitia 4.
Relatia disguise este 1 doar pentru valoarea 2, adica pentru pozitia 3.
Relatia driving este 1 doar pentru valoarea 1, adica pentru pozitia 2.
Relatia germany este 1 doar pentru valoarea 2, adica pentru pozitia 3.
Relatia green este 1 doar pentru valoarea 1, adica pentru pozitia 2.
Relatia james este 1 doar pentru valoarea 3, adica pentru pozitia 4.
Relatia jason este 1 doar pentru valoarea 1, adica pentru pozitia 2.
Relatia libya este 1 doar pentru valoarea 4, adica pentru pozitia 5.
Relatia martialarts este 1 doar pentru valoarea 0, adica pentru pozitia 1.
Relatia parkour este 1 doar pentru valoarea 4, adica pentru pozitia 5.
Relatia pen este 1 doar pentru valoarea 0, adica pentru pozitia 1.
Relatia phone este 1 doar pentru valoarea 3, adica pentru pozitia 4.
Relatia purple este 1 doar pentru valoarea 3, adica pentru pozitia 4.
Relatia red este 1 doar pentru valoarea 0, adica pentru pozitia 1.
Relatia ring este 1 doar pentru valoarea 4, adica pentru pozitia 5.
Relatia russia este 1 doar pentru valoarea 1, adica pentru pozitia 2.
Relatia stan este 1 doar pentru valoarea 0, adica pentru pozitia 1.
Relatia sterling este 1 doar pentru valoarea 4, adica pentru pozitia 5.
Relatia umbrella este 1 doar pentru valoarea 1, adica pentru pozitia 2.

5 Sudoku 4x4

5.1 Cerintele problemei

Sudoku 4x4 este un joc in care se da un patrat de 4x4 pozitii, care are unele pozitii completate cu litere dintre 'a','b','c','d'. Scopul jocului este completarea pozitiiilor astfel incat sa se gaseasca o litera o singura data pe linii, pe coloane si pe fiecare patrat de 2x2 din colturi.

Se cere sa se rezolve urmatorul sudoku 4x4:

			b
b	d		c
	a		

Figure 8: Sudoku 4x4

5.2 Analiza problemei

Pe linia a 2-a lipseste litera 'a', deci aceasta se va plasa pe pozitia (2,3).

In patratul din dreapta sus lipseste litera 'd', deci aceasta se va plasa pe pozitia (1,3).

Pe prima linie lipsesc literele 'a' si 'c'. Litera 'a' nu poate fi pe coloana a 2-a, deoarece aceasta exista deja pe pozitia (4,2), deci vom pune litera 'a' pe pozitia (1,1). Astfel, pe pozitia (1,2) vom pune litera 'c'.

Pe coloana a 2-a lipseste litera 'b', deci aceasta se va plasa pe pozitia (3,2).

Pe coloana a 3-a lipsesc literele 'b' si 'c'. Litera 'b' nu poate fi pe linia a 3-a, deoarece exista deja pe pozitia (3,2), deci vom pune litera 'b' pe pozitia (4,3).

Astfel, pe pozitia (3,3,) vom pune litera 'c'.

In patratul din stanga jos lipsesc literele 'c' si 'd'. Litera 'c' nu poate fi pe linia a 3-a, deoarece exista deja pe pozitia (3,3), deci vom pune litera 'c' pe pozitia (4,1). Astfel, pe pozitia (3,1) vom pune litera 'd'.

Pe linia a 3-a lipseste litera 'a', deci aceasta se va plasa pe pozitia (3,4).

Pe linia a 4-a lipseste litera 'd', deci aceasta se va plasa pe pozitia (4,4).

In concluzie, solutia pentru acest sudoku 4x4 este:

a	c	d	b
b	d	a	c
d	b	c	a
c	a	b	d

Figure 9: Sudoku 4x4 - Solutia

5.3 Implementarea problemei

Cod:

```
1  a11 | a12 | a13 | a14.
2  a21 | a22 | a23 | a24.
3  a31 | a32 | a33 | a34.
4  a41 | a42 | a43 | a44.
5
6  a11 | a21 | a31 | a41.
7  a12 | a22 | a32 | a42.
8  a13 | a23 | a33 | a43.
9  a14 | a24 | a34 | a44.
10
11 b11 | b12 | b13 | b14.
12 b21 | b22 | b23 | b24.
13 b31 | b32 | b33 | b34.
14 b41 | b42 | b43 | b44.
15
16 b11 | b21 | b31 | b41.
17 b12 | b22 | b32 | b42.
18 b13 | b23 | b33 | b43.
19 b14 | b24 | b34 | b44.
20
21 c11 | c12 | c13 | c14.
22 c21 | c22 | c23 | c24.
23 c31 | c32 | c33 | c34.
24 c41 | c42 | c43 | c44.
25
26 c11 | c21 | c31 | c41.
27 c12 | c22 | c32 | c42.
28 c13 | c23 | c33 | c43.
29 c14 | c24 | c34 | c44.
30
31 d11 | d12 | d13 | d14.
32 d21 | d22 | d23 | d24.
33 d31 | d32 | d33 | d34.
34 d41 | d42 | d43 | d44.
35
36 d11 | d21 | d31 | d41.
37 d12 | d22 | d32 | d42.
38 d13 | d23 | d33 | d43.
39 d14 | d24 | d34 | d44.
40
41 a11 -> -a12 & -a13 & -a14.
42 a12 -> -a11 & -a13 & -a14.
43 a13 -> -a12 & -a11 & -a14.
```

```

44  a14 -> -a12 & -a13 & -a11.
45
46  a21 -> -a22 & -a23 & -a24.
47  a22 -> -a21 & -a23 & -a24.
48  a23 -> -a22 & -a21 & -a24.
49  a24 -> -a22 & -a23 & -a21.
50
51  a31 -> -a32 & -a33 & -a34.
52  a32 -> -a31 & -a33 & -a34.
53  a33 -> -a32 & -a31 & -a34.
54  a34 -> -a32 & -a33 & -a31.
55
56  a41 -> -a42 & -a43 & -a44.
57  a42 -> -a41 & -a43 & -a44.
58  a43 -> -a42 & -a41 & -a44.
59  a44 -> -a42 & -a43 & -a41.
60
61  a11 -> -a21 & -a31 & -a41.
62  a21 -> -a11 & -a31 & -a41.
63  a31 -> -a21 & -a11 & -a41.
64  a41 -> -a21 & -a31 & -a11.
65
66  a12 -> -a22 & -a32 & -a42.
67  a22 -> -a12 & -a32 & -a42.
68  a32 -> -a22 & -a12 & -a42.
69  a42 -> -a22 & -a32 & -a12.
70
71  a13 -> -a23 & -a33 & -a43.
72  a23 -> -a13 & -a33 & -a43.
73  a33 -> -a23 & -a13 & -a43.
74  a43 -> -a23 & -a33 & -a13.
75
76  a14 -> -a24 & -a34 & -a44.
77  a24 -> -a14 & -a34 & -a44.
78  a34 -> -a24 & -a14 & -a44.
79  a44 -> -a24 & -a34 & -a14.
80
81  b11 -> -b12 & -b13 & -b14.
82  b12 -> -b11 & -b13 & -b14.
83  b13 -> -b12 & -b11 & -b14.
84  b14 -> -b12 & -b13 & -b11.
85
86  b21 -> -b22 & -b23 & -b24.
87  b22 -> -b21 & -b23 & -b24.
88  b23 -> -b22 & -b21 & -b24.
89  b24 -> -b22 & -b23 & -b21.

```

```

90
91 b31 -> -b32 & -b33 & -b34.
92 b32 -> -b31 & -b33 & -b34.
93 b33 -> -b32 & -b31 & -b34.
94 b34 -> -b32 & -b33 & -b31.
95
96 b41 -> -b42 & -b43 & -b44.
97 b42 -> -b41 & -b43 & -b44.
98 b43 -> -b42 & -b41 & -b44.
99 b44 -> -b42 & -b43 & -b41.
100
101 b11 -> -b21 & -b31 & -b41.
102 b21 -> -b11 & -b31 & -b41.
103 b31 -> -b21 & -b11 & -b41.
104 b41 -> -b21 & -b31 & -b11.
105
106 b12 -> -b22 & -b32 & -b42.
107 b22 -> -b12 & -b32 & -b42.
108 b32 -> -b22 & -b12 & -b42.
109 b42 -> -b22 & -b32 & -b12.
110
111 b13 -> -b23 & -b33 & -b43.
112 b23 -> -b13 & -b33 & -b43.
113 b33 -> -b23 & -b13 & -b43.
114 b43 -> -b23 & -b33 & -b13.
115
116 b14 -> -b24 & -b34 & -b44.
117 b24 -> -b14 & -b34 & -b44.
118 b34 -> -b24 & -b14 & -b44.
119 b44 -> -b24 & -b34 & -b14.
120
121 c11 -> -c12 & -c13 & -c14.
122 c12 -> -c11 & -c13 & -c14.
123 c13 -> -c12 & -c11 & -c14.
124 c14 -> -c12 & -c13 & -c11.
125
126 c21 -> -c22 & -c23 & -c24.
127 c22 -> -c21 & -c23 & -c24.
128 c23 -> -c22 & -c21 & -c24.
129 c24 -> -c22 & -c23 & -c21.
130
131 c31 -> -c32 & -c33 & -c34.
132 c32 -> -c31 & -c33 & -c34.
133 c33 -> -c32 & -c31 & -c34.
134 c34 -> -c32 & -c33 & -c31.
135

```



```

136 c41 -> -c42 & -c43 & -c44.
137 c42 -> -c41 & -c43 & -c44.
138 c43 -> -c42 & -c41 & -c44.
139 c44 -> -c42 & -c43 & -c41.
140
141 c11 -> -c21 & -c31 & -c41.
142 c21 -> -c11 & -c31 & -c41.
143 c31 -> -c21 & -c11 & -c41.
144 c41 -> -c21 & -c31 & -c11.
145
146 c12 -> -c22 & -c32 & -c42.
147 c22 -> -c12 & -c32 & -c42.
148 c32 -> -c22 & -c12 & -c42.
149 c42 -> -c22 & -c32 & -c12.
150
151 c13 -> -c23 & -c33 & -c43.
152 c23 -> -c13 & -c33 & -c43.
153 c33 -> -c23 & -c13 & -c43.
154 c43 -> -c23 & -c33 & -c13.
155
156 c14 -> -c24 & -c34 & -c44.
157 c24 -> -c14 & -c34 & -c44.
158 c34 -> -c24 & -c14 & -c44.
159 c44 -> -c24 & -c34 & -c14.
160
161 d11 -> -d12 & -d13 & -d14.
162 d12 -> -d11 & -d13 & -d14.
163 d13 -> -d12 & -d11 & -d14.
164 d14 -> -d12 & -d13 & -d11.
165
166 d21 -> -d22 & -d23 & -d24.
167 d22 -> -d21 & -d23 & -d24.
168 d23 -> -d22 & -d21 & -d24.
169 d24 -> -d22 & -d23 & -d21.
170
171 d31 -> -d32 & -d33 & -d34.
172 d32 -> -d31 & -d33 & -d34.
173 d33 -> -d32 & -d31 & -d34.
174 d34 -> -d32 & -d33 & -d31.
175
176 d41 -> -d42 & -d43 & -d44.
177 d42 -> -d41 & -d43 & -d44.
178 d43 -> -d42 & -d41 & -d44.
179 d44 -> -d42 & -d43 & -d41.
180
181 d11 -> -d21 & -d31 & -d41.

```

```

182 d21 -> -d11 & -d31 & -d41.
183 d31 -> -d21 & -d11 & -d41.
184 d41 -> -d21 & -d31 & -d11.
185
186 d12 -> -d22 & -d32 & -d42.
187 d22 -> -d12 & -d32 & -d42.
188 d32 -> -d22 & -d12 & -d42.
189 d42 -> -d22 & -d32 & -d12.
190
191 d13 -> -d23 & -d33 & -d43.
192 d23 -> -d13 & -d33 & -d43.
193 d33 -> -d23 & -d13 & -d43.
194 d43 -> -d23 & -d33 & -d13.
195
196 d14 -> -d24 & -d34 & -d44.
197 d24 -> -d14 & -d34 & -d44.
198 d34 -> -d24 & -d14 & -d44.
199 d44 -> -d24 & -d34 & -d14.
200
201 a11 -> -b11 & -c11 & -d11.
202 a12 -> -b12 & -c12 & -d12.
203 a13 -> -b13 & -c13 & -d13.
204 a14 -> -b14 & -c14 & -d14.
205 a21 -> -b21 & -c21 & -d21.
206 a22 -> -b22 & -c22 & -d22.
207 a23 -> -b23 & -c23 & -d23.
208 a24 -> -b24 & -c24 & -d24.
209 a31 -> -b31 & -c31 & -d31.
210 a32 -> -b32 & -c32 & -d32.
211 a33 -> -b33 & -c33 & -d33.
212 a34 -> -b34 & -c34 & -d34.
213 a41 -> -b41 & -c41 & -d41.
214 a42 -> -b42 & -c42 & -d42.
215 a43 -> -b43 & -c43 & -d43.
216 a44 -> -b44 & -c44 & -d44.
217
218 b11 -> -a11 & -c11 & -d11.
219 b12 -> -a12 & -c12 & -d12.
220 b13 -> -a13 & -c13 & -d13.
221 b14 -> -a14 & -c14 & -d14.
222 b21 -> -a21 & -c21 & -d21.
223 b22 -> -a22 & -c22 & -d22.
224 b23 -> -a23 & -c23 & -d23.
225 b24 -> -a24 & -c24 & -d24.
226 b31 -> -a31 & -c31 & -d31.
227 b32 -> -a32 & -c32 & -d32.

```

```

228 b33 -> -a33 & -c33 & -d33.
229 b34 -> -a34 & -c34 & -d34.
230 b41 -> -a41 & -c41 & -d41.
231 b42 -> -a42 & -c42 & -d42.
232 b43 -> -a43 & -c43 & -d43.
233 b44 -> -a44 & -c44 & -d44.
234
235 c11 -> -b11 & -a11 & -d11.
236 c12 -> -b12 & -a12 & -d12.
237 c13 -> -b13 & -a13 & -d13.
238 c14 -> -b14 & -a14 & -d14.
239 c21 -> -b21 & -a21 & -d21.
240 c22 -> -b22 & -a22 & -d22.
241 c23 -> -b23 & -a23 & -d23.
242 c24 -> -b24 & -a24 & -d24.
243 c31 -> -b31 & -a31 & -d31.
244 c32 -> -b32 & -a32 & -d32.
245 c33 -> -b33 & -a33 & -d33.
246 c34 -> -b34 & -a34 & -d34.
247 c41 -> -b41 & -a41 & -d41.
248 c42 -> -b42 & -a42 & -d42.
249 c43 -> -b43 & -a43 & -d43.
250 c44 -> -b44 & -a44 & -d44.
251
252 d11 -> -b11 & -c11 & -a11.
253 d12 -> -b12 & -c12 & -a12.
254 d13 -> -b13 & -c13 & -a13.
255 d14 -> -b14 & -c14 & -a14.
256 d21 -> -b21 & -c21 & -a21.
257 d22 -> -b22 & -c22 & -a22.
258 d23 -> -b23 & -c23 & -a23.
259 d24 -> -b24 & -c24 & -a24.
260 d31 -> -b31 & -c31 & -a31.
261 d32 -> -b32 & -c32 & -a32.
262 d33 -> -b33 & -c33 & -a33.
263 d34 -> -b34 & -c34 & -a34.
264 d41 -> -b41 & -c41 & -a41.
265 d42 -> -b42 & -c42 & -a42.
266 d43 -> -b43 & -c43 & -a43.
267 d44 -> -b44 & -c44 & -a44.
268
269 a11 -> -a22.
270 a12 -> -a21.
271 a21 -> -a12.
272 a22 -> -a11.
273

```

```

274 a13 -> -a24.
275 a14 -> -a23.
276 a24 -> -a13.
277 a23 -> -a14.
278
279 a31 -> -a42.
280 a32 -> -a41.
281 a42 -> -a31.
282 a41 -> -a32.
283
284 a33 -> -a44.
285 a34 -> -a43.
286 a44 -> -a33.
287 a43 -> -a34.
288
289 b11 -> -b22.
290 b12 -> -b21.
291 b21 -> -b12.
292 b22 -> -b11.
293
294 b13 -> -b24.
295 b14 -> -b23.
296 b24 -> -b13.
297 b23 -> -b14.
298
299 b31 -> -b42.
300 b32 -> -b41.
301 b42 -> -b31.
302 b41 -> -b32.
303
304 b33 -> -b44.
305 b34 -> -b43.
306 b44 -> -b33.
307 b43 -> -b34.
308
309 c11 -> -c22.
310 c12 -> -c21.
311 c21 -> -c12.
312 c22 -> -c11.
313
314 c13 -> -c24.
315 c14 -> -c23.
316 c24 -> -c13.
317 c23 -> -c14.
318
319 c31 -> -c42.

```

```

320 c32 -> -c41.
321 c42 -> -c31.
322 c41 -> -c32.
323
324 c33 -> -c44.
325 c34 -> -c43.
326 c44 -> -c33.
327 c43 -> -c34.
328
329 d11 -> -d22.
330 d12 -> -d21.
331 d21 -> -d12.
332 d22 -> -d11.
333
334 d13 -> -d24.
335 d14 -> -d23.
336 d24 -> -d13.
337 d23 -> -d14.
338
339 d31 -> -d42.
340 d32 -> -d41.
341 d42 -> -d31.
342 d41 -> -d32.
343
344 d33 -> -d44.
345 d34 -> -d43.
346 d44 -> -d33.
347 d43 -> -d34.
348
349 b14.
350 b21.
351 d22.
352 c24.
353 a42.

```

5.4 Interpretare rezultate cu Mace4:

```
interpretation( 2, [number = 1,seconds = 0], [  
    relation(a11, [1]),  
    relation(a12, [0]),  
    relation(a13, [0]),  
    relation(a14, [0]),  
    relation(a21, [0]),  
    relation(a22, [0]),  
    relation(a23, [1]),  
    relation(a24, [0]),  
    relation(a31, [0]),  
    relation(a32, [0]),  
    relation(a33, [0]),  
    relation(a34, [1]),  
    relation(a41, [0]),  
    relation(a42, [1]),  
    relation(a43, [0]),  
    relation(a44, [0]),  
  
    relation(b11, [0]),  
    relation(b12, [0]),  
    relation(b13, [0]),  
    relation(b14, [1]),  
    relation(b21, [1]),  
    relation(b22, [0]),  
    relation(b23, [0]),  
    relation(b24, [0]),  
    relation(b31, [0]),  
    relation(b32, [1]),  
    relation(b33, [0]),  
    relation(b34, [0]),  
    relation(b41, [0]),  
    relation(b42, [0]),  
    relation(b43, [1]),  
    relation(b44, [0]),
```

```

relation(c11, [0]),
relation(c12, [1]),
relation(c13, [0]),
relation(c14, [0]),
relation(c21, [0]),
relation(c22, [0]),
relation(c23, [0]),
relation(c24, [1]),
relation(c31, [0]),
relation(c32, [0]),
relation(c33, [1]),
relation(c34, [0]),
relation(c41, [1]),
relation(c42, [0]),
relation(c43, [0]),
relation(c44, [0]),

relation(d11, [0]),
relation(d12, [0]),
relation(d13, [1]),
relation(d14, [0]),
relation(d21, [0]),
relation(d22, [1]),
relation(d23, [0]),
relation(d24, [0]),
relation(d31, [1]),
relation(d32, [0]),
relation(d33, [0]),
relation(d34, [0]),
relation(d41, [0]),
relation(d42, [0]),
relation(d43, [0]),
relation(d44, [1])) .

```

Figure 10: Sudoku 4x4 - interpretare rezultate

Interpretare:

Relatiile a11, a23, a34, a42, b14, b21, b34, b43, c12, c24, c33, c41, d13, d22, d31, d44 au valoarea 1, acestea fiind pozitiile ocupate de caracterele respective. Restul relatiilor au valoarea 0.