Documentație - Puzzle-uri logice

Studenți:
Nicoara Cristian - Cătălin
Tîrziu Cristian - Ioan

Decembrie 2022

Calculatoare și Tehnologia Informației
Facultatea de Autonmatică și Calculatoare
Universitatea Tehnică din Cluj-Napoca

Abstract

În ceea ce urmează, se vor prezenta rezolvările unor puzzle-uri logice.

Keywords—
Books, Books and more Books
Six Clues And Three Houses
Birth Months
Nurse, Accountant and Electrician
Five Friends
Knight and Knaves 39
Knight and Knaves 40
Lady and Tigers – The Ninth Trial
Lady and Tigers – The Tenth Trial
Lady and Tigers – The Eleventh Trial
Lady and Tigers – The Twelfth Trial
Knight and Knaves 51

Contents

1 Books, Books and more Books

Problema poate fi găsită aici.
Enunțul problemei:

> Jake, John, Joe, and Jack each have a different favorite book series. These include "Harry Potter", "The Lord of the Rings", "Sherlock Holmes", and "The Hardy Boys". Using the clues, figure out who likes which book series.
>
> 1. Jake and Joe love mysteries, while John and Jack prefer magic and fantasy.
>
> 2. "Joe" is the name of a character in Joe's favorite series.
>
> 3. John's Halloween costume, based on his favorite series, includes glasses and a black robe.

Figure 1: Books, Books and more Books - enunț

Am abordat această problemă astfel:

Am definit elementele distincte care apar în problemă, urmând să scriem propozițiile logice, ținând cont de indiciile date în problemă.

Se rulează în terminal următoarea comandă: mace4 -f books.in

În urma rulării, s-a obținut următorul model:

```
============================== MODEL =================================

interpretation( 4, [number=1, seconds=0], [

        function(harrypotter, [ 0 ]),

        function(jack, [ 1 ]),

        function(jake, [ 2 ]),

        function(joe, [ 3 ]),

        function(john, [ 0 ]),

        function(lordoftherings, [ 1 ]),

        function(sherlockholmes, [ 2 ]),

        function(thehardyboys, [ 3 ])
]).

============================== end of model ==========================

============================== STATISTICS ============================

For domain size 4.

Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=24, kept=24.
Selections=3, assignments=6, propagations=5, current_models=1.
Rewrite_terms=62, rewrite_bools=30, indexes=0.
Rules_from_neg_clauses=2, cross_offs=12.

============================== end of statistics =====================

User_CPU=0.00, System_CPU=0.00, Wall_clock=0.

Exiting with 1 model.
```

Figure 2: Books, Books and more Books - terminal

Conform modelului, cartea preferată a lui John este "Harry Potter" din: funtion(john, [0]) și funtion(harrypotter, [0]), cartea preferată a lui Jack este "The Lord of the Rings" din: funtion(jack, [1]) și funtion(lordoftherings, [1]), cartea preferată a lui Jake este "Sherlock Holmes" din: funtion(jake, [2]) și funtion(sherlockholmes, [2]), iar cartea preferată a lui Joe este "The Hardy Boys" din: funtion(joe, [3]) și funtion(harrypotter, [3]).

Modelul găsit coincide cu soluția oferită pe site.

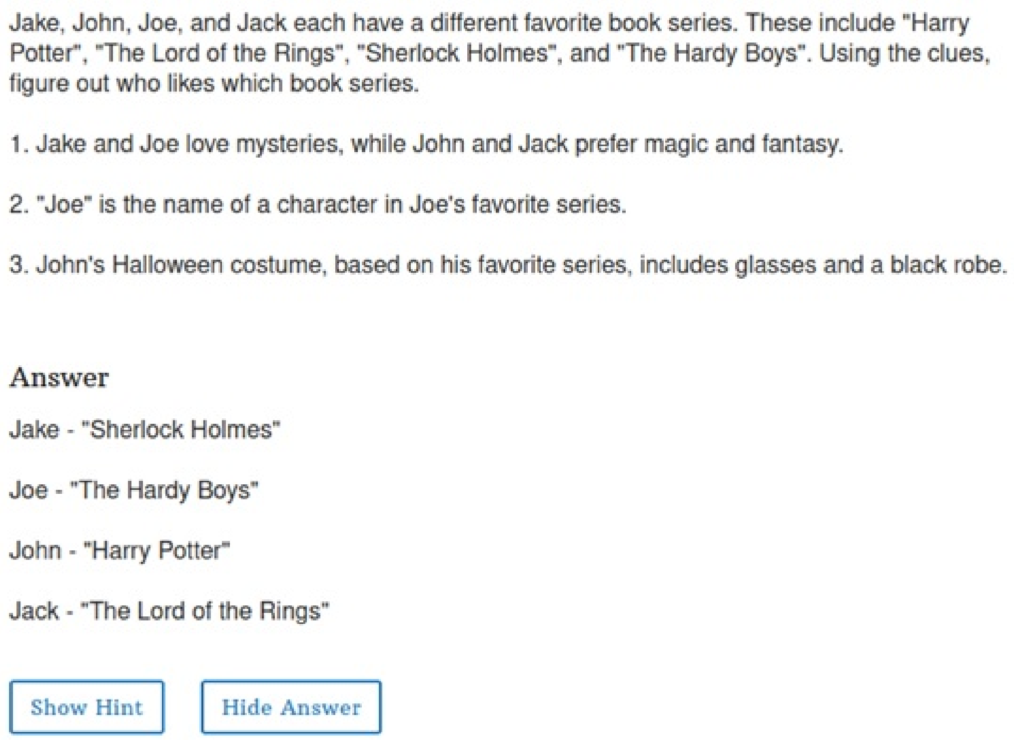


Figure 3: Books, Books and more Books - soluție

Rezolvarea propusă de noi se poate găsi în fișierul "books.in". De asemenea, se poate găsi și în partea de Appendix.

2 Six Clues And Three Houses

Problema poate fi găsită aici.

Enunțul problemei:

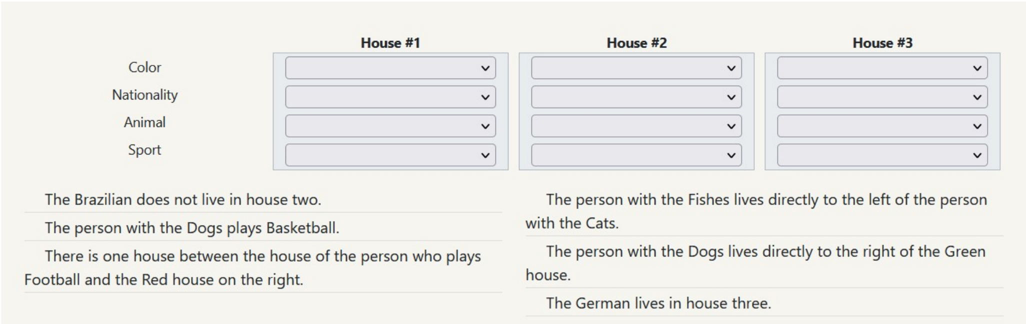


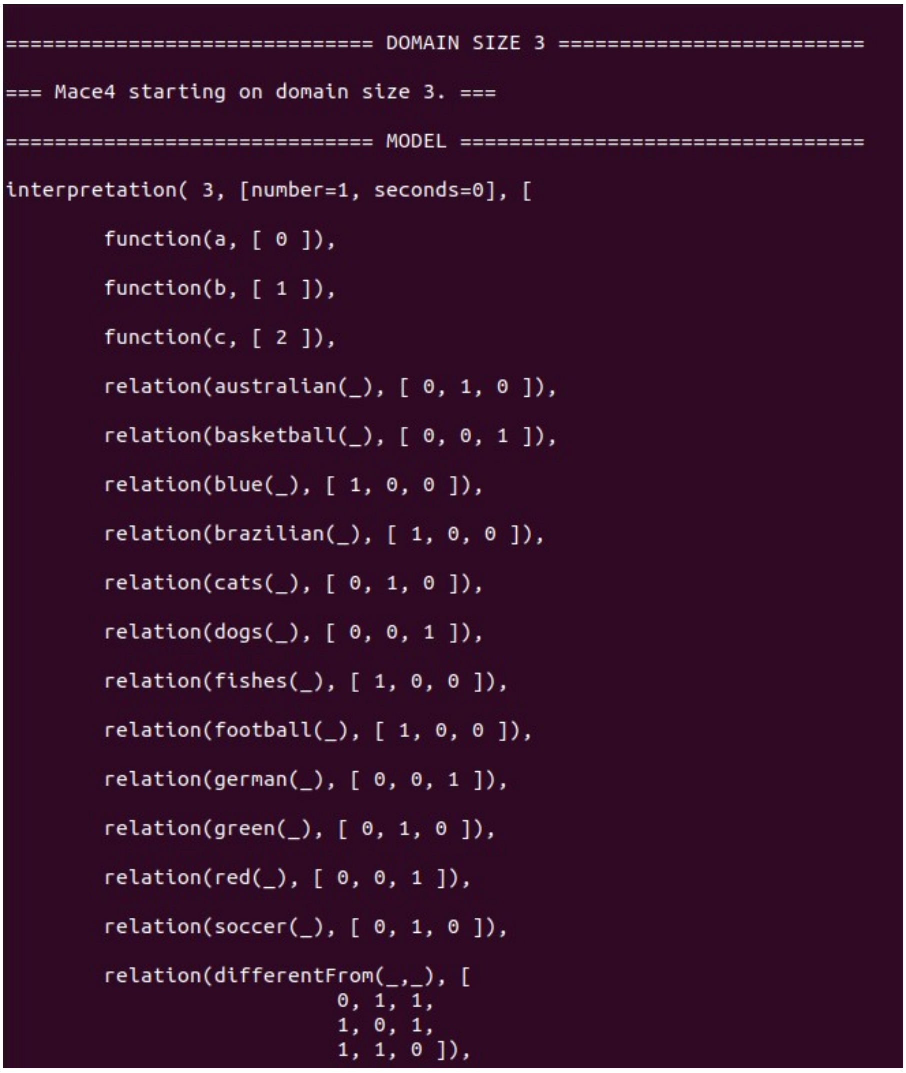
Figure 4: Six Clues And Three Houses - enunț

Am abordat această problemă astfel:

Ne-am folosit de funcția differentFrom pentru a menționa că sunt trei entități diferite de aflat. Folosindu-ne de funcția rightNeighbour am definit care pereche de entități se învecinează și care nu. A urmat scrierea propozițiilor logice, având în vedere indiciile menționate.

Se rulează în terminal următoarea comandă: mace4 -f sixCluesThreeHouses.in

În urma rulării, s-a obținut următorul model:



```
============================== DOMAIN SIZE 3 =========================

=== Mace4 starting on domain size 3. ===

============================== MODEL =================================

interpretation( 3, [number=1, seconds=0], [

        function(a, [ 0 ]),

        function(b, [ 1 ]),

        function(c, [ 2 ]),

        relation(australian(_), [ 0, 1, 0 ]),

        relation(basketball(_), [ 0, 0, 1 ]),

        relation(blue(_), [ 1, 0, 0 ]),

        relation(brazilian(_), [ 1, 0, 0 ]),

        relation(cats(_), [ 0, 1, 0 ]),

        relation(dogs(_), [ 0, 0, 1 ]),

        relation(fishes(_), [ 1, 0, 0 ]),

        relation(football(_), [ 1, 0, 0 ]),

        relation(german(_), [ 0, 0, 1 ]),

        relation(green(_), [ 0, 1, 0 ]),

        relation(red(_), [ 0, 0, 1 ]),

        relation(soccer(_), [ 0, 1, 0 ]),

        relation(differentFrom(_,_), [
                   0, 1, 1,
                   1, 0, 1,
                   1, 1, 0 ]),
```

Figure 5: Six Clues And Three Houses - terminal - prima parte

```
        relation(german(_), [ 0, 0, 1 ]),

        relation(green(_), [ 0, 1, 0 ]),

        relation(red(_), [ 0, 0, 1 ]),

        relation(soccer(_), [ 0, 1, 0 ]),

        relation(differentFrom(_,_), [
                            0, 1, 1,
                            1, 0, 1,
                            1, 1, 0 ]),

        relation(neighbor(_,_), [
                            0, 1, 0,
                            1, 0, 1,
                            0, 1, 0 ]),

        relation(rightNeighbor(_,_), [
                            0, 1, 0,
                            0, 0, 1,
                            0, 0, 0 ])
]).

============================== end of model ==========================

============================== STATISTICS ============================

For domain size 3.

Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=221, kept=221.
Selections=7, assignments=13, propagations=126, current_models=1.
Rewrite_terms=33, rewrite_bools=688, indexes=15.
Rules_from_neg_clauses=1, cross_offs=3.

============================== end of statistics =====================

User_CPU=0.00, System_CPU=0.00, Wall_clock=0.

Exiting with 1 model.
```

Figure 6: Six Clues And Three Houses - terminal - a doua parte

Conform modelului, se observa ca australianul este in a doua casă, jucătorul de basket în cea de-a treia, prima casă este albastră, brazilianul se află în prima casă, proprietarul de pisici în cea de-a doua casă, proprietarul de câini în cea de-a treia casă, proprietarul de pești în prima casă, jucătorul de fotbal în prima casă, germanul în a treia casă, a doua casă este verde, a treia este roșie, iar jucătorul de soccer este în a doua casă.

Astfel aflăm urmatoarele:

Casa 1	Casa 2	Casa 3
albastră	verde	roșie
brazilian	australian	german
pești	pisici	câini
fotbal	soccer	basket

Table 1: Six Clues And Three Houses

Modelul găsit coincide cu soluția oferită pe site.

Figure 7: Six Clues And Three Houses - soluție

Rezolvarea propusă de noi se poate găsi în fișierul "sixCluesThreeHouses.in". De asemenea, se poate găsi și în partea de Appendix.

3 Birth Months

Problema poate fi găsită aici.

Enunțul problemei:

Four sisters, Sara, Ophelia, Nora, and Dawn, were each born in a different one of the months September, October, November, and December.

"This is terrible," said Ophelia one day. "None of us have an initial that matches the initial of her birth month."

"I don't mind at all," replied the girl who was born in September.

"That's easy for you to say," said Nora. "It would at least be cool if the initial of my birth month was a vowel, but no."

In which month was each girl born?

Figure 8: Birth Months - enunț

Am abordat această problemă astfel:

Am definit elementele distincte care apar în problemă, urmând să scriem propozițiile logice, ținând cont de indiciile date în problemă.

Se rulează în terminal următoarea comandă: mace4 -f birthMonths.in

În urma rulării, s-a obținut următorul model:

```
============================== DOMAIN SIZE 4 =========================

=== Mace4 starting on domain size 4. ===

============================== MODEL =================================

interpretation( 4, [number=1, seconds=0], [

        function(dawn, [ 0 ]),

        function(december, [ 1 ]),

        function(nora, [ 1 ]),

        function(november, [ 2 ]),

        function(october, [ 3 ]),

        function(ophelia, [ 2 ]),

        function(sara, [ 3 ]),

        function(september, [ 0 ])
]).

============================== end of model ==========================

============================== STATISTICS ============================

For domain size 4.

Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=23, kept=23.
Selections=5, assignments=13, propagations=8, current_models=1.
Rewrite_terms=104, rewrite_bools=47, indexes=0.
Rules_from_neg_clauses=7, cross_offs=27.

============================== end of statistics =====================

User_CPU=0.00, System_CPU=0.00, Wall_clock=0.

Exiting with 1 model.
```

Figure 9: Birth Months - terminal

Conform modelului, se observă că Dawn este născută în luna septembrie, Nora în luna Decembrie, Ophelia în luna noiembrie, iar Sara în luna octombrie.

Modelul găsit coincide cu soluția oferită pe site.

Four sisters, Sara, Ophelia, Nora, and Dawn, were each born in a different one of the months September, October, November, and December.

"This is terrible," said Ophelia one day. "None of us have an initial that matches the initial of her birth month."

"I don't mind at all," replied the girl who was born in September.

"That's easy for you to say," said Nora. "It would at least be cool if the initial of my birth month was a vowel, but no."

In which month was each girl born?

Answer

Sara was born in October, Ophelia was born in November, Nora was born in December, and Dawn was born in September.

Nora's birth month obviously cannot be November. Since it doesn't start with a vowel, it cannot be October. Since she was talking to the girl who was born in September, it cannot be September. Therefore, it must be December.

The girl born in September obviously cannot be Sara. Since she was talking to both Ophelia and Nora, it cannot be either of them. Therefore, it must be Dawn.

This leaves the girls Sara and Ophelia and the months October and November. Ophelia cannot be born in October, so she must be born in November, and Sara was born in October.

Figure 10: Birth Months - soluție

Rezolvarea propusă de noi se poate găsi în fișierul "birthMonths.in". De asemenea, se poate găsi și în partea de Appendix.

4 Nurse, Accountant and Electrician

Problema poate fi găsită aici.

Enunțul problemei:

> Nicky, Jared, and Antoine are an electrician, accountant and nurse. Jared is the electrician's brother and Antoine has never met the electrician or the accountant. Which person has each job?

Figure 11: Nurse, Accountant and Electrician - enunț

Am abordat această problemă astfel:

Am definit elementele distincte care apar în problemă, urmând să scriem propozițiile logice, ținând cont de indiciile date în problemă.

Se rulează în terminal următoarea comandă: mace4 -f nurseAccountantElectrician.in

În urma rulării, s-a obținut următorul model:

```
============================== MODEL =================================

interpretation( 3, [number=1, seconds=0], [

        function(accountant, [ 0 ]),

        function(antoine, [ 1 ]),

        function(electrician, [ 2 ]),

        function(jared, [ 0 ]),

        function(nicky, [ 2 ]),

        function(nurse, [ 1 ])
]).

============================== end of model ==========================

============================== STATISTICS ============================

For domain size 3.

Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=9, kept=9.
Selections=2, assignments=3, propagations=4, current_models=1.
Rewrite_terms=21, rewrite_bools=10, indexes=0.
Rules_from_neg_clauses=4, cross_offs=11.

============================== end of statistics =====================

User_CPU=0.00, System_CPU=0.00, Wall_clock=0.

Exiting with 1 model.
```

Figure 12: Nurse, Accountant and Electrician - terminal

Conform modelului, Jared este contabil, Antoine este asistent medical, iar Nicky este electrician.

Modelul găsit coincide cu soluția oferită pe site.

Nicky, Jared, and Antoine are an electrician, accountant and nurse. Jared is the electrician's brother and Antoine has never met the electrician or the accountant. Which person has each job?

Answer

Antoine is the nurse. Jared is the accountant. Nicky is the electrician.

Figure 13: Nurse, Accountant and Electrician - soluție

Rezolvarea propusă de noi se poate găsi în fișierul ”nurseAccountantElectrician.in”. De asemenea, se poate găsi și în partea de Appendix.

5 Five Friends

Problema poate fi găsită aici.

Enunțul problemei:

There are 5 friends going to a movie (Chris, Joey, Andy, Kelsey, & Tina). How are they standing in line for a ticket?

Chris is in front of Andy.
Kelsey is behind Andy.
Tina is behind Joey.
Joey is in front of Chris.
Tina is in front of Andy.
Tina is behind Chris.

Figure 14: Five Friends - enunț

Am abordat această problemă astfel:

Am definit elementele distincte care apar în problemă, am definit funcția front, pentru a decide cine se află în fața altei persoane și cine nu. A urmat să scriem propozițiile logice, ținând cont de indiciile date în problemă.

Se rulează în terminal următoarea comandă: mace4 -f fiveFriends.in

În urma rulării, s-a obținut următorul model:

```
============================== MODEL =================================

interpretation( 5, [number=1, seconds=0], [

        function(a, [ 0 ]),

        function(andy, [ 3 ]),

        function(b, [ 1 ]),

        function(c, [ 2 ]),

        function(chris, [ 1 ]),

        function(d, [ 3 ]),

        function(e, [ 4 ]),

        function(joey, [ 0 ]),

        function(kelsey, [ 4 ]),

        function(tina, [ 2 ]),

        relation(front(_,_), [
                           0, 1, 1, 1, 1,
                           0, 0, 1, 1, 1,
                           0, 0, 0, 1, 1,
                           0, 0, 0, 0, 1,
                           0, 0, 0, 0, 0 ])
]).

============================== end of model ==========================

============================== STATISTICS ============================

For domain size 5.

Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=71, kept=71.
Selections=7, assignments=25, propagations=41, current_models=1.
Rewrite_terms=240, rewrite_bools=140, indexes=20.
Rules_from_neg_clauses=16, cross_offs=78.

============================== end of statistics =====================

User_CPU=0.00, System_CPU=0.00, Wall_clock=0.

Exiting with 1 model.
```

Figure 15: Five Friends - terminal

Conform modelului, ordinea prietenilor este: Joey, Chris, Tina, Andy, Kelsey.
Modelul găsit coincide cu soluția oferită pe site.

There are 5 friends going to a movie (Chris, Joey, Andy, Kelsey, & Tina). How are they standing in line for a ticket?

Chris is in front of Andy.
Kelsey is behind Andy.
Tina is behind Joey.
Joey is in front of Chris.
Tina is in front of Andy.
Tina is behind Chris.

Answer

Joey, Chris, Tina, Andy, Kelsey

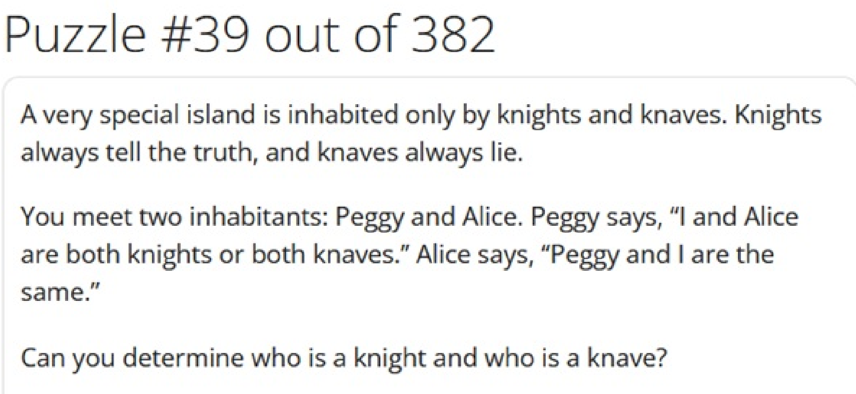
Figure 16: Five Friends - soluție

Rezolvarea propusă de noi se poate găsi în fișierul "fiveFriends.in". De asemenea, se poate găsi și în partea de Appendix.

6 Knight and Knaves 39

Problema poate fi găsită aici.

Enunțul problemei:



Puzzle #39 out of 382

A very special island is inhabited only by knights and knaves. Knights always tell the truth, and knaves always lie.

You meet two inhabitants: Peggy and Alice. Peggy says, “I and Alice are both knights or both knaves.” Alice says, “Peggy and I are the same.”

Can you determine who is a knight and who is a knave?

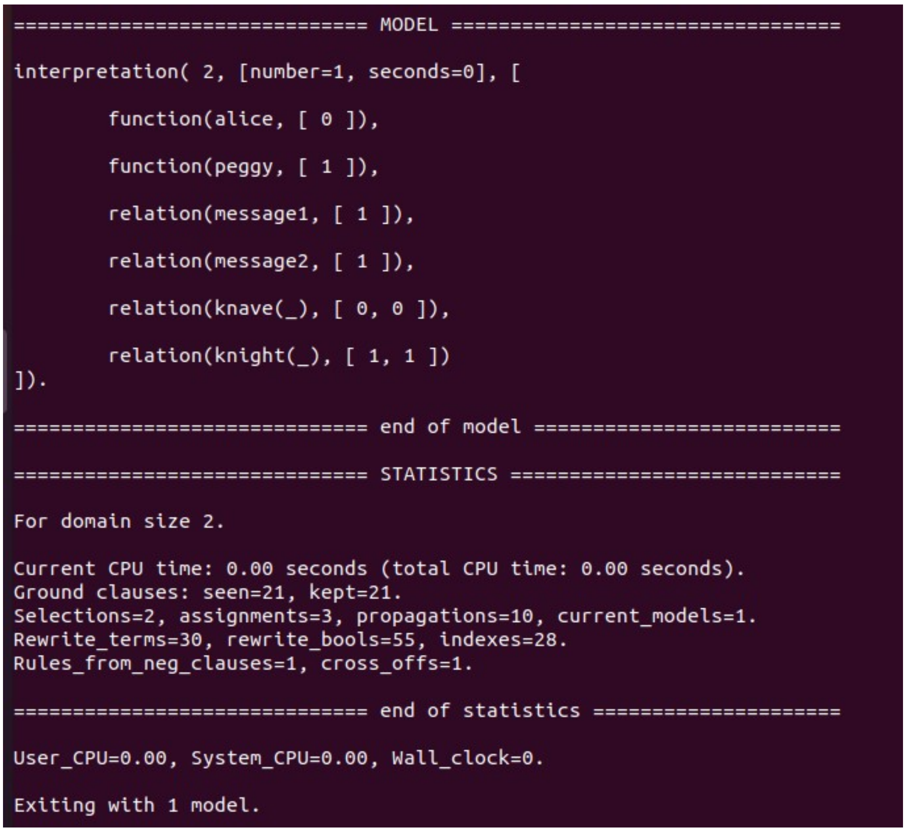
Figure 17: Knight and Knaves 39 - enunț

Am abordat această problemă astfel:

Am definit elementele distincte care apar în problemă, am difinit mesajele primite si am scris propozițiile logice, ținând cont de indiciile date în problemă.

Se rulează în terminal următoarea comandă: mace4 -f knightsAndKnaves39.in

În urma rulării, s-a obținut următorul model:



```
============================== MODEL =================================

interpretation( 2, [number=1, seconds=0], [

        function(alice, [ 0 ]),

        function(peggy, [ 1 ]),

        relation(message1, [ 1 ]),

        relation(message2, [ 1 ]),

        relation(knave(_), [ 0, 0 ]),

        relation(knight(_), [ 1, 1 ])
]).

============================== end of model ==========================

============================== STATISTICS ============================

For domain size 2.

Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=21, kept=21.
Selections=2, assignments=3, propagations=10, current_models=1.
Rewrite_terms=30, rewrite_bools=55, indexes=28.
Rules_from_neg_clauses=1, cross_offs=1.

============================== end of statistics =====================

User_CPU=0.00, System_CPU=0.00, Wall_clock=0.

Exiting with 1 model.
```

Figure 18: Knight and Knaves 39 - terminal

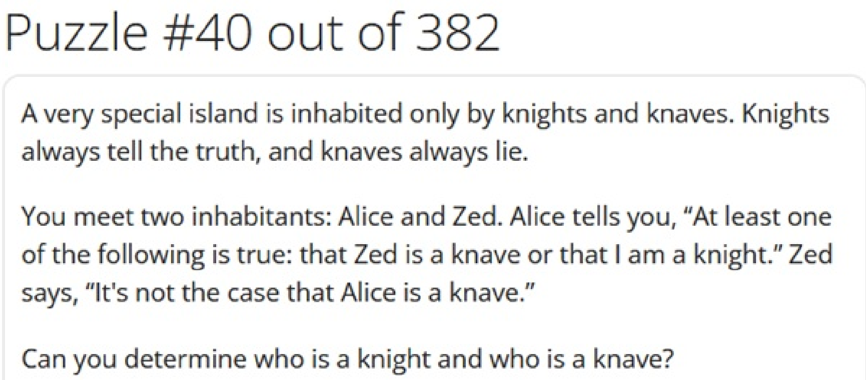
Conform modelului, atât Peggy, cât și Alice sunt Knight.

Rezolvarea propusă de noi se poate găsi în fișierul ”knightsAndKnaves39.in”. De asemenea, se poate găsi și în partea de Appendix.

7 Knight and Knaves 40

Problema poate fi găsită aici.

Enunțul problemei:



Puzzle #40 out of 382

A very special island is inhabited only by knights and knaves. Knights always tell the truth, and knaves always lie.

You meet two inhabitants: Alice and Zed. Alice tells you, “At least one of the following is true: that Zed is a knave or that I am a knight.” Zed says, “It's not the case that Alice is a knave.”

Can you determine who is a knight and who is a knave?

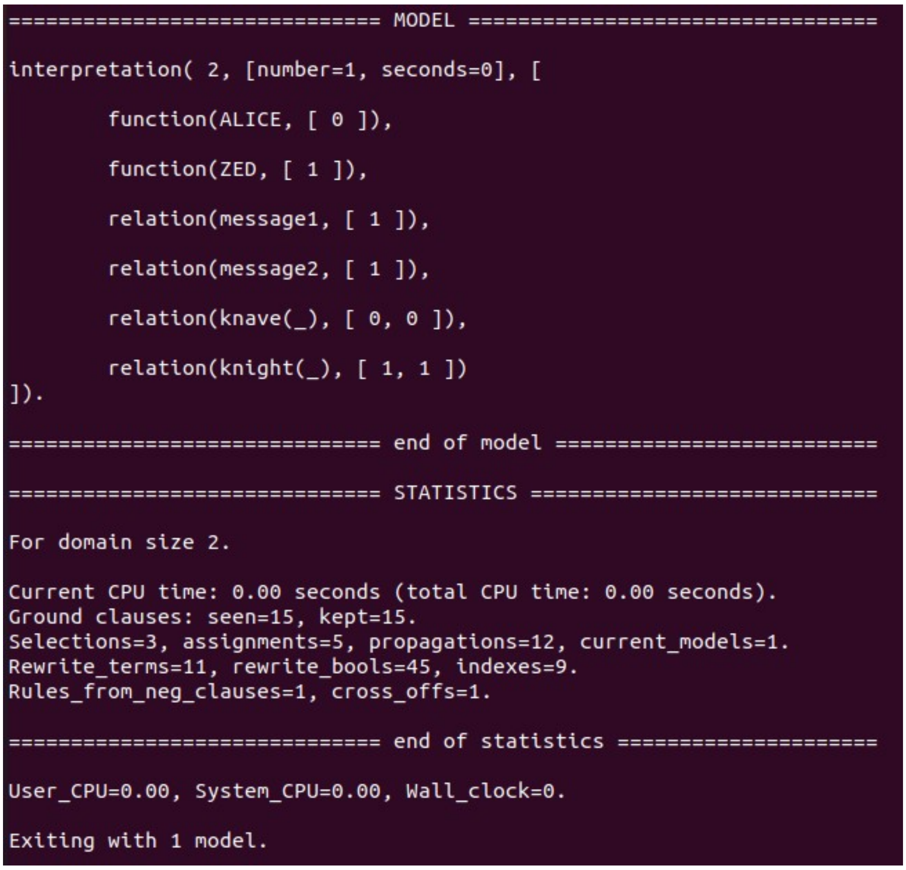
Figure 19: Knight and Knaves 40 - enunț

Am abordat această problemă astfel:

Am definit elementele distincte care apar în problemă, am difinit mesajele primite si am scris propozițiile logice, ținând cont de indiciile date în problemă.

Se rulează în terminal următoarea comandă: mace4 -f knightsAndKnaves40.in

În urma rulării, s-a obținut următorul model:



```
============================== MODEL =================================

interpretation( 2, [number=1, seconds=0], [

        function(ALICE, [ 0 ]),

        function(ZED, [ 1 ]),

        relation(message1, [ 1 ]),

        relation(message2, [ 1 ]),

        relation(knave(_), [ 0, 0 ]),

        relation(knight(_), [ 1, 1 ])
]).

============================== end of model ==========================

============================== STATISTICS ============================

For domain size 2.

Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=15, kept=15.
Selections=3, assignments=5, propagations=12, current_models=1.
Rewrite_terms=11, rewrite_bools=45, indexes=9.
Rules_from_neg_clauses=1, cross_offs=1.

============================== end of statistics =====================

User_CPU=0.00, System_CPU=0.00, Wall_clock=0.

Exiting with 1 model.
```

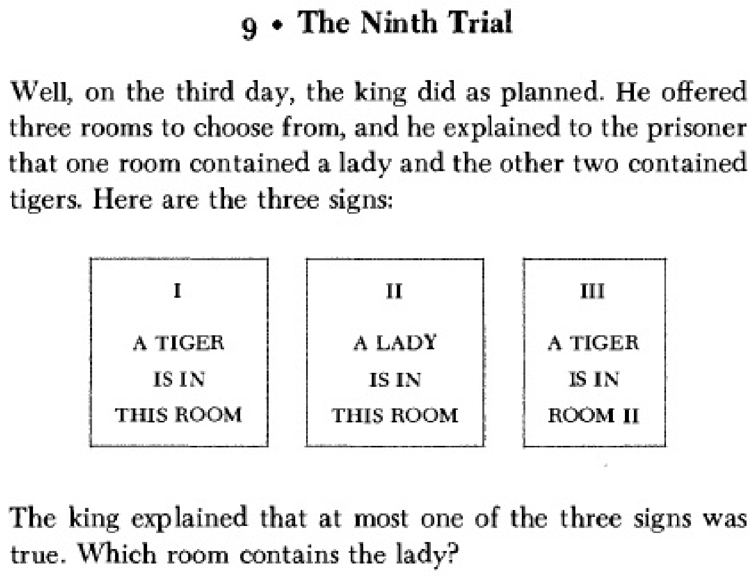
Figure 20: Knight and Knaves 40 - terminal

Conform modelului, atât Alice, cât și Zed sunt Knight.

Rezolvarea propusă de noi se poate găsi în fișierul ”knightsAndKnaves40.in”. De asemenea, se poate găsi și în partea de Appendix.

8 Lady and Tigers – The Ninth Trial

Enunțul problemei:



9 • The Ninth Trial

Well, on the third day, the king did as planned. He offered three rooms to choose from, and he explained to the prisoner that one room contained a lady and the other two contained tigers. Here are the three signs:

I	II	III
A TIGER IS IN THIS ROOM	A LADY IS IN THIS ROOM	A TIGER IS IN ROOM II

The king explained that at most one of the three signs was true. Which room contains the lady?

Figure 21: Lady and Tigers – The Ninth Trial - enunt

Am abordat această problemă astfel:
Am scris propoziții logice care să ne asigure că există o prințesă într-una dintre camere, că există o singură prințesă, că în celelalte camere se află tigri și că în fiecare cameră cu tigri există doar un tigru, iar în camera cu prințesa este doar prințesa. A urmat să mai adăugăm propoziții logice care să țină cont de indiciile din enunțul problemei.

Se rulează în terminal următoarea comandă: mace4 -f ladyAndTigers9.in

În urma rulării, s-a obținut următorul model:

```
============================== MODEL =================================

interpretation( 2, [number=1, seconds=0], [

        relation(l1, [ 1 ]),

        relation(l2, [ 0 ]),

        relation(l3, [ 0 ]),

        relation(t1, [ 0 ]),

        relation(t2, [ 1 ]),

        relation(t3, [ 1 ])
]).

============================== end of model ==========================

============================== STATISTICS ============================

For domain size 2.

Current CPU time: 0.00 seconds (total CPU time: 0.01 seconds).
Ground clauses: seen=16, kept=16.
Selections=2, assignments=4, propagations=12, current_models=1.
Rewrite_terms=0, rewrite_bools=53, indexes=0.
Rules_from_neg_clauses=0, cross_offs=0.

============================== end of statistics =====================

User_CPU=0.01, System_CPU=0.00, Wall_clock=0.

Exiting with 1 model.
```

Figure 22: Lady and Tigers – The Ninth Trial - terminal

Conform modelului, prințesa se află în prima cameră, iar în celelalte camere sunt tigri.
Modelul găsit coincide cu soluția oferită.

9 • Signs II and III contradict each other, so at least one of them is true. Since at most one of the three signs is true, then the first one must be false, so the lady is in Room I.

Figure 23: Lady and Tigers – The Ninth Trial - soluție

Rezolvarea propusă de noi se poate găsi în fișierul "ladyAndTigers9.in". De asemenea, se poate găsi și în partea de Appendix.

9 Lady and Tigers – The Tenth Trial

Enunțul problemei:

10 • The Tenth Trial

Again there was only one lady and two tigers. The king explained to the prisoner that the sign on the door of the room containing the lady was true, and that at least one of the other two signs was false.

Here are the signs:

LADIES OR TIGERS?

I	II	III
A TIGER IS IN ROOM II	A TIGER IS IN THIS ROOM	A TIGER IS IN ROOM I

What should the prisoner do?

Figure 24: Lady and Tigers – The Tenth Trial - enunț

Am abordat această problemă asemănător problemei precedente.
Se rulează în terminal următoarea comandă: mace4 -f ladyAndTigers10.in
În urma rulării, s-a obținut următorul model:

```
============================== MODEL =================================

interpretation( 2, [number=1, seconds=0], [

        relation(l1, [ 1 ]),

        relation(l2, [ 0 ]),

        relation(l3, [ 0 ]),

        relation(t1, [ 0 ]),

        relation(t2, [ 1 ]),

        relation(t3, [ 1 ])
]).

============================== end of model ==========================

============================== STATISTICS ============================

For domain size 2.

Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=19, kept=19.
Selections=2, assignments=4, propagations=10, current_models=1.
Rewrite_terms=0, rewrite_bools=50, indexes=0.
Rules_from_neg_clauses=0, cross_offs=0.

============================== end of statistics =====================

User_CPU=0.00, System_CPU=0.00, Wall_clock=0.

Exiting with 1 model.
```

Figure 25: Lady and Tigers – The Tenth Trial - terminal

Conform modelului, prințesa se află în prima cameră, iar în celelalte camere sunt tigri.

Modelul găsit coincide cu soluția oferită.

10 • Since the sign of the room containing the lady is true, then the lady certainly can't be in Room II. If she is in Room III, then all three signs must be true, which is contrary to the given condition that at least one sign is false. Therefore, the lady is in Room I (and sign II is true and sign III is false).

Figure 26: Lady and Tigers – The Tenth Trial - soluție

Rezolvarea propusă de noi se poate găsi în fișierul ”ladyAndTigers10.in”. De asemenea, se poate găsi și în partea de Appendix.

10 Lady and Tigers – The Eleventh Trial

Enunțul problemei:

11 • First, Second, and Third Choice

In this more whimsical trial, the king explained to the prisoner that one of the three rooms contained a lady, another a tiger, and the third room was empty. The sign on the door of the room containing the lady was true, the sign on the door of the room with the tiger was false, and the sign on the door of the empty room could be either true or false. Here are the signs:

I	II	III
ROOM III IS EMPTY	THE TIGER IS IN ROOM I	THIS ROOM IS EMPTY

Now, the prisoner happened to know the lady in question and wished to marry her. Therefore, although the empty room was preferable to the one with the tiger, his first choice was the room with the lady.

Which room contains the lady, and which room contains the tiger? If you can answer these two questions, you should have little difficulty in also determining which room is empty.

Figure 27: Lady and Tigers – The Eleventh Trial - enunț

Am abordat această problemă asemanător cu problemele precedente cu prințesa și tigri, dar, conform problemei, am adăugat o nouă condiție - aceea că una dintre camere este goală.

Se rulează în terminal următoarea comandă: mace4 -f ladyAndTigers11.in

În urma rulării, s-a obținut următorul model:

```
============================== MODEL =================================

interpretation( 2, [number=1, seconds=0], [

        relation(e1, [ 0 ]),

        relation(e2, [ 0 ]),

        relation(e3, [ 1 ]),

        relation(l1, [ 1 ]),

        relation(l2, [ 0 ]),

        relation(l3, [ 0 ]),

        relation(t1, [ 0 ]),

        relation(t2, [ 1 ]),

        relation(t3, [ 0 ])
]).

============================== end of model ==========================

============================== STATISTICS ============================

For domain size 2.

Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=24, kept=24.
Selections=2, assignments=2, propagations=7, current_models=1.
Rewrite_terms=0, rewrite_bools=35, indexes=0.
Rules_from_neg_clauses=0, cross_offs=0.

============================== end of statistics =====================

User_CPU=0.00, System_CPU=0.00, Wall_clock=0.

Exiting with 1 model.
```

Figure 28: Lady and Tigers – The Eleventh Trial - terminal

Conform modelului, a treia cameră este goală, prințesa este în prima cameră, iar tigrul în cea de-a doua. Modelul găsit coincide cu soluția oferită.

11 ◆ Since the sign on the door of the room containing the lady is true, then the lady cannot be in Room III.

Suppose she is in Room II. Then sign II would be true; hence the tiger would be in Room I and Room III would be empty. This would mean that the sign on the door of the tiger's room would be true, which is not possible. Therefore, the lady is in Room I; Room III must then be empty, and the tiger is in Room II.

Figure 29: Lady and Tigers – The Eleventh Trial - soluție

Rezolvarea propusă de noi se poate găsi în fișierul ”ladyAndTigers11.in”. De asemenea, se poate găsi și în partea de Appendix.

11 Lady and Tigers – The Twelfth Trial

Enunțul problemei:

12 • A Logical Labyrinth

Well, the king was as good as his word. Instead of having three rooms for the prisoner to choose from, he gave him nine! As he explained, only one room contained a lady; each of the other eight either contained a tiger or was empty. And, the king added, the sign on the door of the room containing the lady is true; the signs on doors of all rooms containing tigers are false; and the signs on doors of empty rooms can be either true or false.

Here are the signs:

I THE LADY IS IN AN ODD-NUMBERED ROOM	II THIS ROOM IS EMPTY	III EITHER SIGN V IS RIGHT OR SIGN VII IS WRONG
IV SIGN I IS WRONG	V EITHER SIGN II OR SIGN IV IS RIGHT	VI SIGN III IS WRONG
VII THE LADY IS NOT IN ROOM I	VIII THIS ROOM CONTAINS A TIGER AND ROOM IX IS EMPTY	IX THIS ROOM CONTAINS A TIGER AND VI IS WRONG

The prisoner studied the situation for a long while.

"The problem is unsolvable!" he exclaimed angrily. "That's not fair!"

"I know," laughed the king.

"Very funny!" replied the prisoner. "Come on, now, at least give me a decent clue: is Room Eight empty or not?"

The king was decent enough to tell him whether Room VIII was empty or not, and the prisoner was then able to deduce where the lady was.

Which room contained the lady?

Figure 30: Lady and Tigers – The Twelfth Trial - enunț

Am abordat această problemă astfel:

În mod asemănător celor trei probleme anterioare, am scris propoziții logice, luând în calcul faptu că, pentru această problemă există 9 uși. Am scris propoziții logice despre fiecare indiciu de pe fiecare ușă.

Se rulează în terminal următoarea comandă: mace4 -f ladyAndTigers12.in

În urma rulării, s-a obținut următorul model:

```
============================== DOMAIN SIZE 2 =========================

=== Mace4 starting on domain size 2. ===

============================== MODEL =================================

interpretation( 2, [number=1, seconds=0], [

        relation(e1, [ 0 ]),

        relation(e2, [ 0 ]),

        relation(e3, [ 0 ]),

        relation(e4, [ 0 ]),

        relation(e5, [ 0 ]),

        relation(e6, [ 0 ]),

        relation(e7, [ 0 ]),

        relation(e8, [ 0 ]),

        relation(e9, [ 1 ]),

        relation(l1, [ 0 ]),

        relation(l2, [ 0 ]),

        relation(l3, [ 0 ]),

        relation(l4, [ 0 ]),

        relation(l5, [ 0 ]),

        relation(l6, [ 0 ]),

        relation(l7, [ 1 ]),

        relation(l8, [ 0 ]),

        relation(l9, [ 0 ]),
```

Figure 31: Lady and Tigers – The Twelfth Trial - terminal - partea I

```
        relation(l8, [ 0 ]),

        relation(l9, [ 0 ]),

        relation(t1, [ 0 ]),

        relation(t2, [ 0 ]),

        relation(t3, [ 0 ]),

        relation(t4, [ 0 ]),

        relation(t5, [ 1 ]),

        relation(t6, [ 0 ]),

        relation(t7, [ 0 ]),

        relation(t8, [ 0 ]),

        relation(t9, [ 0 ])
]).

============================== end of model ==========================

============================== STATISTICS ============================

For domain size 2.

Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=132, kept=130.
Selections=14, assignments=14, propagations=13, current_models=1.
Rewrite_terms=0, rewrite_bools=185, indexes=0.
Rules_from_neg_clauses=0, cross_offs=0.

============================== end of statistics =====================

User_CPU=0.00, System_CPU=0.01, Wall_clock=0.

Exiting with 1 model.
```

Figure 32: Lady and Tigers – The Twelfth Trial - terminal - partea a II-a

Conform modelului, prințesa se află în camera cu numărul șapte.
Modelul găsit coincide cu soluția oferită.

12 • If the king had told the prisoner that Room VIII was empty, it would have been impossible for the prisoner to have found the lady. Since the prisoner did deduce where the lady was, the king must have told him that Room VIII was not empty, and the prisoner reasoned as follows:

Figure 33: Lady and Tigers – The Twelfth Trial - soluție

"It is impossible for the lady to be in Room Eight, for if she were, Sign Eight would be true, but the sign says a tiger is in the room, which would be a contradiction. Therefore, Room Eight does not contain the lady. Also, Room Eight is not empty; therefore, Room Eight must contain a tiger. Since it contains a tiger, the sign is false. Now, if Room Nine is empty, then Sign Eight would be true; therefore, Room Nine cannot be empty.

"So, Room Nine is also not empty. It cannot contain the lady, or the sign would be true, which would mean that the room contains a tiger; this means Sign Nine is false. If Sign Six is really wrong, then Sign Nine would be true, which is not possible. Therefore, Sign Six is right.

"Since Sign Six is right, then Sign Three is wrong. The only way Sign Three can be wrong is that Sign Five is Wrong and Sign Seven is right. Since Sign Five is wrong, then Sign Two and Sign Four are both wrong. Since Sign Four is wrong, then Sign One must be right.

"Now I know which signs are right and which signs are wrong—namely:

"1 – Right	4 – Wrong	7 – Right
2 – Wrong	5 – Wrong	8 – Wrong
3 – Wrong	6 – Right	9 – Wrong

"I now know that the lady is in either Room One, Room Six, or Room Seven, since the others all have false signs. Since Sign One is right, the lady can't be in Room Six. Since Sign Seven is right, the lady can't be in Room One. Therefore, the lady is in Room Seven."

Figure 34: Lady and Tigers – The Twelfth Trial - soluție

Rezolvarea propusă de noi se poate găsi în fișierul "ladyAndTigers12.in". De asemenea, se poate găsi și în partea de Appendix.

12 Knight and Knaves 51

Problema poate fi găsită aici.

Enunțul problemei:

Puzzle #51 out of 382

A very special island is inhabited only by knights and knaves. Knights always tell the truth, and knaves always lie.

You meet three inhabitants: Alice, Rex and Bob. Alice tells you that Rex is a knave. Rex tells you that it's false that Bob is a knave. Bob claims, "I am a knight or Alice is a knight."

Can you determine who is a knight and who is a knave?

Figure 35: Knight and Knaves 51 - enunț

Am abordat această problemă astfel:

Am definit elementele distincte care apar în problemă, am difinit mesajele primite și am scris propozițiile logice, ținând cont de indiciile date în problemă.

Se rulează în terminal următoarea comandă: mace4 -f knightsAndKnaves51.in

În urma rulării, s-a obținut următorul model:

```
============================== MODEL =================================

interpretation( 3, [number=1, seconds=0], [

        function(alice, [ 0 ]),

        function(bob, [ 1 ]),

        function(rex, [ 2 ]),

        relation(message1, [ 0 ]),

        relation(message2, [ 1 ]),

        relation(message3, [ 1 ]),

        relation(knave(_), [ 1, 0, 0 ]),

        relation(knight(_), [ 0, 1, 1 ])
]).

============================== end of model ==========================

============================== STATISTICS ============================

For domain size 3.

Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=27, kept=27.
Selections=3, assignments=5, propagations=15, current_models=1.
Rewrite_terms=24, rewrite_bools=57, indexes=16.
Rules_from_neg_clauses=1, cross_offs=3.

============================== end of statistics =====================

User_CPU=0.00, System_CPU=0.00, Wall_clock=0.

Exiting with 1 model.
```

Figure 36: Knight and Knaves 51 - terminal

Conform modelului, se observă că Aice este Knave, iar Bob și Rex sunt Knight.

Rezolvarea propusă de noi se poate găsi în fișierul "knightsAndKnaves51.in". De asemenea, se poate găsi și în partea de Appendix.

References

https://www.braingle.com/brainteasers/category.php?category=Logic&fields=&value=&unrated=0;order=8
https://www.brainzilla.com/logic/zebra/
https://philosophy.hku.hk/think/logic/knights.php

Appendix

books.in

```
assign(domain_size, 4).
assign(max_models, -1).

list(distinct).
    [jake, john, joe, jack].
    [harrypotter, sherlockholmes, thehardyboys, lordoftherings].
end_of_list.

formulas(assumptions).


(jack = harrypotter | jack = lordoftherings) &
(john = harrypotter | john = lordoftherings) &
(jake = sherlockholmes | jake = thehardyboys) &
(joe = thehardyboys | joe = sherlockholmes).

(joe = thehardyboys & joe != sherlockholmes) &
(jake = sherlockholmes & jake != thehardyboys).

(john = harrypotter & john != lordoftherings) &
(jack = lordoftherings & jack != harrypotter).

end_of_list.

formulas(goals).
end_of_list.
```

sixCluesThreeHouses.in

```
assign(domain_size, 3).
assign(max_models, -1).

formulas(assumptions).

differentFrom(a,b).
differentFrom(a,c).
differentFrom(b,c).

differentFrom(x,y) -> differentFrom(y,x).

rightNeighbor(a,b).
rightNeighbor(b,c).
-rightNeighbor(a,a).
-rightNeighbor(a,c).
-rightNeighbor(b,a).
-rightNeighbor(b,b).
-rightNeighbor(c,a).
-rightNeighbor(c,b).
-rightNeighbor(c,c).

rightNeighbor(x,y) | rightNeighbor(y,x) <-> neighbor(x,y).

blue(x) | green(x) | red(x).
brazilian(x) | australian(x) | german(x).
fishes(x) | cats(x) | dogs(x).
football(x) | soccer(x) | basketball(x).

red(x) & red(y) -> -differentFrom(x,y).
blue(x) & blue(y) -> -differentFrom(x,y).
green(x) & green(y) -> -differentFrom(x,y).
```

```
brazilian(x) & brazilian(y) -> -differentFrom(x,y).
australian(x) & australian(y) -> -differentFrom(x,y).
german(x) & german(y) -> -differentFrom(x,y).

fishes(x) & fishes(y) -> -differentFrom(x,y).
cats(x) & cats(y) -> -differentFrom(x,y).
dogs(x) & dogs(y) -> -differentFrom(x,y).

football(x) & football(y) -> -differentFrom(x,y).
soccer(x) & soccer(y) -> -differentFrom(x,y).
basketball(x) & basketball(y) -> -differentFrom(x,y).

german(c).
-brazilian(b).
dogs(x) <-> basketball(x).
cats(x) & fishes(y) -> rightNeighbor(y,x).
dogs(x) & green(y) -> rightNeighbor(y,x).
football(x) & red(y) -> -neighbor(x,y).
green(x) & red(y) -> -rightNeighbor(y,x).
blue(x) & red(y) -> -rightNeighbor(y,x).

end_of_list.

formulas(goals).
end_of_list.
```

birthMonths.in

```
assign(domain_size, 4).
assign(max_models, -1).

list(distinct).
    [sara, ophelia, nora, dawn].
    [september, october, november, december].
end_of_list.

formulas(assumptions).

(sara = october) | (sara = november) |(sara = december).
(ophelia = september) | (ophelia = november) |(ophelia = december).
(nora = september) | (nora = october) |(nora = december).
(dawn = september) | (dawn = october) |(dawn = november).

sara != september.
ophelia != october.
nora != november.
dawn != december.
nora != october.
nora != september.
ophelia != september.


end_of_list.

formulas(goals).
end_of_list.
```

nurseAccountantElectrician.in

```
assign(domain_size, 3).
assign(max_models, -1).

list(distinct).
    [nicky, jared, antoine].
    [nurse, accountant, electrician].
end_of_list.

formulas(assumptions).

jared != electrician.
antoine != electrician.
antoine != accountant.


end_of_list.

formulas(goals).
end_of_list.
```

fiveFriends.in

```
assign(domain_size, 5).
assign(max_models, -1).

list(distinct).
    [chris, joey, andy, kelsey, tina].
end_of_list.

formulas(assumptions).

front(a,b).
front(a,c).
front(a,d).
front(a,e).
front(b,c).
front(b,d).
front(b,e).
front(c,d).
front(c,e).
front(d,e).
-front(a,a).
-front(b,a).
-front(c,a).
-front(d,a).
-front(e,a).
-front(b,b).
-front(c,b).
-front(d,b).
-front(e,b).
-front(c,c).
-front(d,c).
-front(e,c).
-front(d,d).
-front(e,d).
-front(e,e).

front(x,y) -> -front(y,x).

front(chris,andy) & front(andy,kelsey) & front(joey,tina) &
front(joey,chris) & front(tina,andy) & front(chris,tina).
-front(chris,chris).
-front(andy,andy).
```

```
-front(kelsey,kelsey).
-front(joey,joey).
-front(tina,tina).

end_of_list.

formulas(goals).
end_of_list.
```

knightsAndKnaves39.in

```
assign(domain_size, 2).
assign(max_models, -1).

list(distinct).
    [peggy, alice].
end_of_list.

formulas(assumptions).


message1 -> (knight(peggy) & knight(alice)) | (knave(peggy) & knave(alice)).
message2 -> (knight(peggy) & knight(alice)) | (knave(peggy) & knave(alice)).

knight(peggy) <-> knight(alice).


knight(x) <-> -knave(x).


knight(peggy) -> message1.
knave(peggy) -> -message1.

knight(alice) -> message2.
knave(alice) -> -message2.

knave(peggy) -> knight(alice).
knave(alice) -> knight(peggy).

end_of_list.

formulas(goals).
end_of_list.
```

knightsAndKnaves40.in

```
assign(domain_size, 2).
assign(max_models, -1).

list(distinct).
    [ALICE, ZED].
end_of_list.

formulas(assumptions).

knave(x) <-> -knight(x).
knight(x) <-> -knave(x).

%knave(ALICE) | knight(ALICE).
%knave(ZED) | knight(ZED).


message1 -> knave(ZED) | knight(ALICE).
message2 -> -knave(ALICE).
```

```
%knave(ALICE) -> -message1  .
 knight(ALICE) -> message1.

 knight(ZED) -> message2.
%knave(ZED) -> -message2.

 knave(ALICE) -> knight(ZED).
 knave(ZED) -> knave(ALICE).

 end_of_list.

 formulas(goals).
 end_of_list.
```

ladyAndTigers9.in

```
 assign(max_seconds, 30).

%set(binary_resolution).
%set(print_gen).

 formulas(assumptions).
%there is a lady in room 1, 2, or 3
l1 | l2 | l3.
%no lady in more than 1 room; the princess is unique
     l1 -> -l2.
     l1 -> -l3.
     l2 -> -l1.
     l2 -> -l3.
     l3 -> -l1.
     l3 -> -l2.

%there are 2 tigers in two of the room 1, 2, or 3
 t1 & t2 | t2 & t3 | t1 & t3.

%no tiger in the room where the lady stays
l1 -> -t1.
l2 -> -t2.
l3 -> -t3.

 (t1 & -l2 & -t2) | (-t1 & l2 & -t2) | (-t1 & -l2 & t2) | (-t1 & -l2 & -t2).

 end_of_list.

 formulas(goals).
 end_of_list.
```

ladyAndTigers10.in

```
 assign(max_seconds, 30).

%set(binary_resolution).
%set(print_gen).

 formulas(assumptions).
%there is a lady in room 1, 2, or 3
l1 | l2 | l3.
%no lady in more than 1 room; the princess is unique
     l1 -> -l2.
     l1 -> -l3.
     l2 -> -l1.
     l2 -> -l3.
     l3 -> -l1.
```

```
    l3 -> -l2.

%there are 2 tigers in two of the room 1, 2, or 3
t1 & t2 | t2 & t3 | t1 & t3.

%no tiger in the room where the lady stays
l1 -> -t1.
l2 -> -t2.
l3 -> -t3.

%clue on door #1: there is a tiger in room #2
l1 -> t2.

%clue on door #2: there is a tiger in room #2
l2 -> t2.

%clue on door #3: there is a tiger in room #1
l3 -> t1.

%at least one of the clues on tiger rooms lies.
(t1 & t2) -> (-t2 | -t2).
(t2 & t3) -> (-t2 | -t1).
(t1 & t3) -> (-t2 | -t1).

end_of_list.

formulas(goals).
end_of_list.
```

ladyAndTigers11.in

```
assign(max_seconds, 30).

%set(binary_resolution).
%set(print_gen).

formulas(assumptions).
%there is a lady in room 1, 2, or 3
l1 | l2 | l3.
%no lady in more than 1 room; the princess is unique
    l1 -> -l2.
    l1 -> -l3.
    l2 -> -l1.
    l2 -> -l3.
    l3 -> -l1.
    l3 -> -l2.

%there is one tiger in room 1, 2, or 3
t1 | t2 | t3.

%there one empty room
e1 | e2 | e3.

%no tiger and emptiness in the room where the lady stays
l1 -> -t1 & -e1.
l2 -> -t2 & -e2.
l3 -> -t3 & -e3.

%no emptiness in the room where the tiger stays
t1 -> -e1.
t2 -> -e2.
t3 -> -e3.


%clue on door #1: the room #3 is empty
```

```
l1 -> e3.
t1 -> -e3.
e1 -> e3 | -e3.

%clue on door #2: there is a tiger in room #1
l2 -> t1.
t2 -> -t1.
e2 -> t1 | -t1.

%clue on door #3: this room is empty
l3 -> e3.
t3 -> -e3.
e3 -> e3 | -e3.

end_of_list.

formulas(goals).
end_of_list.
```

ladyAndTigers12.in

```
%set(binary_resolution).
%set(print_gen).

formulas(assumptions).

l1 | l2 | l3 | l4 | l5 | l6 | l7 | l8 | l9.

   l1 -> -l2.      l2 -> -l1.      l3 -> -l1.      l4 -> -l1.
   l1 -> -l3.      l2 -> -l3.      l3 -> -l2.      l4 -> -l2.
   l1 -> -l4.      l2 -> -l4.      l3 -> -l4.      l4 -> -l3.
   l1 -> -l5.      l2 -> -l5.      l3 -> -l5.      l4 -> -l5.
   l1 -> -l6.      l2 -> -l6.      l3 -> -l6.      l4 -> -l6.
   l1 -> -l7.      l2 -> -l7.      l3 -> -l7.      l4 -> -l7.
   l1 -> -l8.      l2 -> -l8.      l3 -> -l8.      l4 -> -l8.
   l1 -> -l9.      l2 -> -l9.      l3 -> -l9.      l4 -> -l9.

   l5 -> -l1.      l6 -> -l1.      l7 -> -l1.      l8 -> -l1.
   l5 -> -l2.      l6 -> -l2.      l7 -> -l2.      l8 -> -l2.
   l5 -> -l3.      l6 -> -l3.      l7 -> -l3.      l8 -> -l3.
   l5 -> -l4.      l6 -> -l4.      l7 -> -l4.      l8 -> -l4.
   l5 -> -l6.      l6 -> -l5.      l7 -> -l5.      l8 -> -l5.
   l5 -> -l7.      l6 -> -l7.      l7 -> -l6.      l8 -> -l6.
   l5 -> -l8.      l6 -> -l8.      l7 -> -l8.      l8 -> -l7.
   l5 -> -l9.      l6 -> -l9.      l7 -> -l9.      l8 -> -l9.

   l9 -> -l1.
   l9 -> -l2.
   l9 -> -l3.
   l9 -> -l4.
   l9 -> -l5.
   l9 -> -l6.
   l9 -> -l7.
   l9 -> -l8.

t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9.

e1 | e2 | e3 | e4 | e5 | e6 | e7 | e8 | e9.

l1 -> -t1 & -e1.
l2 -> -t2 & -e2.
l3 -> -t3 & -e3.
l4 -> -t4 & -e4.
l5 -> -t5 & -e5.
```

```
l6 -> -t6 & -e6.
l7 -> -t7 & -e7.
l8 -> -t8 & -e8.
l9 -> -t9 & -e9.

t1 -> -e1.
t2 -> -e2.
t3 -> -e3.
t4 -> -e4.
t5 -> -e5.
t6 -> -e6.
t7 -> -e7.
t8 -> -e8.
t9 -> -e9.


%clue on door #1
l1 -> l1 | l3 | l5 | l7 | l9.
t1 -> l2 | l4 | l6 | l8.
e1 -> l1 | l2 | l3 | l4 | l5 | l6 | l7 | l8 | l9.

%clue on door #2
l2 -> e2.
t2 -> -e2.
e2 -> e2 | -e2.

%clue on door #3
l3 -> (e2 | (l2 | l4 | l6 | l8)) | l1.
t3 -> (-e2 | (l1 | l3 | l5 | l7 | l9)) | -l1.
e3 -> (e2 | (l2 | l4 | l6 | l8) | l1) |
      ((-e2 | (l1 | l3 | l5 | l7 | l9)) | -l1).

%clue on door #4
l4 -> l2 | l4 | l6 | l8.
t4 -> l1 | l3 | l5 | l7 | l9.
e4 -> -(l1 | l2 | l3 | l4 | l5 | l6 | l7 | l8 | l9).

%clue on door #5
l5 -> e2 | (l2 | l4 | l6 | l8).
t5 -> -e2 | (l1 | l3 | l5 | l7 | l9).
e5 -> e2 | -e2.

%clue on door #6
l6 -> -((e2 | (l2 | l4 | l6 | l8)) | l1).
t6 -> (e2 | (l2 | l4 | l6 | l8)) | l1.
e6 -> (e2 | (l2 | l4 | l6 | l8) | l1) |
      ((-e2 | (l1 | l3 | l5 | l7 | l9)) | -l1).

%clue on door #7
l7 -> -l1.
t7 -> l1.
e7 -> l1 | -l1.

%clue on door #8
l8 -> t8 & e9.
t8 -> -t8 & -e9.
e8 -> (t8 & e9) | (-t8 & -e9).

%clue on door #9
l9 -> t9 & ((e2 | (l2 | l4 | l6 | l8)) | l1).
t9 -> -t9 & ((-e2 | (l1 | l3 | l5 | l7 | l9)) | -l1).
e9 -> (t9 & ((e2 | (l2 | l4 | l6 | l8)) | l1)) |
      (-t9 & ((-e2 | (l1 | l3 | l5 | l7 | l9)) | -l1)).


end_of_list.
```

```
formulas(goals).
end_of_list.
```

knightsAndKnaves51.in

```
assign(domain_size, 3).
assign(max_models, -1).

list(distinct).
    [alice, rex, bob].
end_of_list.

formulas(assumptions).

knave(x) <-> -knight(x).
knight(x) <-> -knave(x).

message1 -> knave(rex).
message2 -> knight(bob).
message3 -> knight(bob) | knight(alice).

knave(alice) -> -message1 .
knight(alice) -> message1.

knight(rex) -> message2.
knave(rex) -> -message2.

knight(bob) -> message3.
knave(bob) -> -message3.

knave(alice) -> knight(rex).
knave(rex) -> knave(bob).
knave(bob) -> knave(alice).

end_of_list.

formulas(goals).
end_of_list.
```