Artificial intelligence - Project 2
- Logical Agents -

Vesa Bianca

8/12/2020

1 Problem 1

1.1 Tattoos Puzzle

In this section the solution for the following problem will be presented:

Maurice had several customers in his tattoo parlor today, each of which requested a simple tattoo of their astrological sign.

Match each customer to their zodiac sign and tattoo color, and determine the price for each.

1. The Sagittarius paid 5 dollars less than the Cancer.
2. Of the customer who paid 55 dollars and Isaac, one was the Taurus and the other got the orange tattoo.
3. The Aquarius was either the person who got the red tattoo or the customer who paid 45 dollars.
4. Mario paid 5 dollars less than Sheila.
5. The Pisces paid 25 dollars more than the person who got the green tattoo.
6. Neil was either the customer who paid 55 dollars or the customer who got the violet tattoo.
7. Of the person who got the green tattoo and the customer who got the black tattoo, one was the Sagittarius and the other was Isaac.
8. The person who paid 65 dollars was either the customer who got the violet tattoo or the customer who got the blue tattoo.
9. The Taurus paid 5 dollars less than Mario.
10. Zachary paid 25 dollars less than the customer who got the blue tattoo.
12. The Virgo paid 10 dollars less than the customer who got the pink tattoo.
13. Kendra didn't get the violet tattoo.
14. The customer who got the green tattoo paid 5 dollars more than Zachary.
15. Neil didn't pay 65 dollars.

1.1.1 Code implementation

Code:

```
assign(domain_size, 7).
set(arithmetic).

list(distinct).

        [Thirtyfive, Fourty, Fourtyfive, Fifty, Fiftyfive, Sixty, Sixtyfive].
        [Isaac, Janie, Kendra, Mario, Neil, Sheila, Zachary].
        [Black, Blue, Green, Pink, Orange, Red, Violet].
        [Aquarius, Cancer, Pisces, Libra, Sagittarius, Taurus, Virgo].

end_of_list.

formulas(assumption).

        Thirtyfive = 0.
        Fourty = 1.
        Fourtyfive = 2.
        Fifty = 3.
        Fiftyfive = 4.
        Sixty = 5.
        Sixtyfive = 6.

        Cancer = Sagittarius + 1.
```

```
24          ( (Taurus = 4 ) & (Isaac = Orange) ) | ( (Orange = 4) & (Isaac = Taurus) ).
25          (Aquarius = Red) | (Aquarius = 2).
26          Sheila = Mario + 1.
27          Pisces = Green + 5.
28          (Neil = 4) | (Neil = Violet).
29          ( (Green = Sagittarius ) & (Black = Isaac ) ) | ( (Green = Isaac ) & (Black = Sagittarius ) ).
30          (Violet = 6) | (Blue = 6).
31          Mario = Taurus + 1.
32          Blue = Zachary + 5.
33          Pink = Virgo + 2.
34          Kendra != Violet.
35          Green = Zachary + 1.
36          Neil != 6.
37
38  end_of_list.
```

Explanation:

- For this problem I came up with the following implementation:
- In the first lines **4 to 11**, I defined four lists of distinct constants, that in the end have to be mapped one to one from each list.
- Because most of the clues are given around the prices of the tattoos, I decided to give a correspondant number to each price (lines **15 to 21**, in ascending order from the domain size.
- The prices given, have an increment of 5, so in order to be able to work with them, each time we have to increment a value with a certain price, we will divide it by 5, so it will fit in the domain size.
- Every clue is translated in first order logic in lines **23 to 36**.
- Clue 1 can be written as Cancer is 5 dollars more that Sagittarius, and five will be divided by 5, so 1 will be added.
- Clue 2 means either Taurus is worth 55 dollars and Isaac got the Orange tattoo, or the Orange tattoo was worth 55 dollars and Isaac got the Taurus.
- Clue 3 translates into: either the Aquarius tattoo was Red, or worth 45 dollars.
- Clue 4 means Sheila's tattoo was worth Mario's plus 5 dollars (divided by 5).
- Clue 5 tells us that the Pisces tattoo was worth as much as the Green one plus 25 dollars (divided by 5).
- Clue 6 is easily put into propositional logic.
- From clue 7 we know that either the Sagittarius tattoo is Green and Isaac's is Black, or the Sagittarius tattoo is Black and Isaac's is Green.
- Clue 8 says that either the Violet tattoo was worth 65 dollars or the Blue one was worth the same sum.
- With clue 9 we can figure out that Mario's tattoo was worth 5 dollars more (divided by 5) than the Taurus one.
- Clue 10 says that the Blue tattoo was worth as much as Zachary's plus 25 (divided by 5).
- Clue 11 reveals that the Pink one's price is the Virgo one's plus 10 (divided by 5).
- Clue 12 is easily translatable in line 34.
- Clue 13 means that the Green tattoo is Zachary's plus 5 (divided by 5).
- Clue 14 is written clearly in line 36.

Commands:

- mace4 -m -1 -f tattoos.in

1.1.2 Solution Explanation and Intermediate Steps

This sub-section is dedicated to detailing the logical solution of the problem together with some additional steps that can be taken, in order to reach the final goal. Each step can be used to showcase the fact that the logic has been implemented correctly. All the intermediate goals built up the final one.

- From clues 2 and 7 we can figure out that Isaac can't have the Orange tattoo, since he also have the Green or Black one, therefore Isaac will have the Taurus tattoo. This assumption (Isaac = Taurus) can be an intermediate step for which mace finds possible models.
- With the information that we just deduced, the orange tattoo is worth 55 dollars from clue 2, so Orange = 4 is another true assumption.
- If 65 dollars is either violet or blue then SixtyFive != Black and SixtyFive != Green and SixtyFive != Pink and SixtyFive != Orange and SixtyFive != Red.
- If Aquarius is either Red or FourtyFive then Red != FourtyFive.
- If Sagittarius is 1 step less than Cancer, then Cancer != ThirtyFive.
- If Mario is 1 step less than Sheila, then Mario != SixtyFive.
- If Mario is 1 step less than Sheila, then Sheila != ThirtyFive.
- If Virgo is less than pink by some specific amount, then Virgo != Pink.
- If Virgo is 2 steps less than pink, then Virgo != Sixty, Virgo != SixtyFive, Virgo != FourtyFive, Pink != ThirtyFive and Pink != Fourty.
- If Pisces is 5 steps greater than green, then Green != FourtyFive, Green != Fifty, Green != Sixty, Pisces != ThirtyFive, Pisces != Fourty, Pisces != FourtyFive, Pisces != Fifty, Pisces != FiftyFive.
- If Taurus is 1 step less than Mario, then Taurus != SixtyFive, Mario != ThirtyFive.
- If Mario is 1 step less than Sheila, and Mario != ThirtyFive, then Sheila != Fourty.
- If Taurus is 1 step less than Mario, and Mario != SixtyFive, then Taurus != Sixty.
- If green is 1 step greater than Zachary, then Zachary != SixtyFive, Green != ThirtyFive.
- Since all other possibilities have been eliminated for the price of the green tattoo, it means that Green = Fourty, which means the Pisces = SixtyFive (clue 5), Zachary = ThirtyFive (clue 13) and Blue = Sixty (clue 10) .
- From clue 8 we can say that SixtyFive = Violet.
- With clue 6 combined with clue 14, and the information above, we have the assumption that NEil = FiftyFive.
- FourtyFive = Pink, since all other possibilities have been eliminated, and with clue 11 Virgo = ThirtyFive.
- ThirtyFive = Red, since all other possibilities have been eliminated, and alongside with clue 3, Aquarius = FourtyFive.
- Fifty = Black, since all other possibilities have been eliminated.
- We know that Isaac = Taurus and the only spots without a person and a zodiac sign are Fourty, Fifty and Sixty. If Isaac would equal Sixty, then Mario would equal SixtyFive, which leaves no room for Sheila, who has paid 5 more dollars than Mario. The same reasoning applies if we put Isaac on FourtyFive. Therefore, Isaac = 40, Mario = FourtyFive and Sheila = Fifty.
- Kendra can't be Violet so Kendra = Sixty, Sagittarius = Fifty and Cancer = FiftyFive.
- In the end, all that is left is Libra = Sixty and Janie = SixtyFive.

1.1.3 Personal observations and notes

Because this puzzle has only one correct solution, I chose to represent it with mace4, which helped me obtain one final model that fits the given clues.

2 Problem 2

2.1 River Crossing Problem

In this section the solution for the following problem will be presented:

As a wildfire rages through the grass lands 3 lions and 3 wildebeest flee for their lives.
To escape the inferno, they must cross over the left bank of a river.
Fortunately, there happens to be a raft nearby, that can carry up to two animals at a time and needs at least one animal on board to guide it across the river.
But if the lions ever outnumber the wildebeest on either side of the river, the lions will kill the wildebeest.
What is the fastest way for all 6 animals to get across, without the lions killing the wildebeest?

2.1.1 Code implementation

Code:

```
formulas(assumptions).

 %right(x, y) means animal x is at the right side of the river, at moment y
 %left(x, y) means animal x is at the left side of the river, at moment y
 right(W1, 0).
 right(W2, 0).
 right(W3, 0).
 right(L1, 0).
 right(L2, 0).
 right(L3, 0).

 % one wildebeest and two lions on the same bank
 right(W1, x) & right(L1, x) & right(L2, x) & left(W2, x) & left(W3, x) & left(L3, x) -> eaten.
 right(W1, x) & right(L1, x) & right(L3, x) & left(W2, x) & left(W3, x) & left(L2, x) -> eaten.
 right(W1, x) & right(L2, x) & right(L3, x) & left(W2, x) & left(W3, x) & left(L1, x) -> eaten.
 left(W1, x) & left(L1, x) & left(L2, x) & right(W2, x) & right(W3, x) & right(L3, x) -> eaten.
 left(W1, x) & left(L1, x) & left(L3, x) & right(W2, x) & right(W3, x) & right(L2, x) -> eaten.
 left(W1, x) & left(L2, x) & left(L3, x) & right(W2, x) & right(W3, x) & right(L1, x) -> eaten.

 right(W2, x) & right(L1, x) & right(L2, x) & left(W1, x) & left(W3, x) & left(L3, x) -> eaten.
 right(W2, x) & right(L1, x) & right(L3, x) & left(W1, x) & left(W3, x) & left(L2, x) -> eaten.
 right(W2, x) & right(L2, x) & right(L3, x) & left(W1, x) & left(W3, x) & left(L1, x) -> eaten.
 left(W2, x) & left(L1, x) & left(L2, x) & right(W1, x) & right(W3, x) & right(L3, x) -> eaten.
 left(W2, x) & left(L1, x) & left(L3, x) & right(W1, x) & right(W3, x) & right(L2, x) -> eaten.
 left(W2, x) & left(L2, x) & left(L3, x) & right(W1, x) & right(W3, x) & right(L1, x) -> eaten.

 right(W3, x) & right(L1, x) & right(L2, x) & left(W1, x) & left(W2, x) & left(L3, x) -> eaten.
 right(W3, x) & right(L1, x) & right(L3, x) & left(W1, x) & left(W2, x) & left(L2, x) -> eaten.
 right(W3, x) & right(L2, x) & right(L3, x) & left(W1, x) & left(W2, x) & left(L1, x) -> eaten.
 left(W3, x) & left(L1, x) & left(L2, x) & right(W1, x) & right(W2, x) & right(L3, x) -> eaten.
 left(W3, x) & left(L1, x) & left(L3, x) & right(W1, x) & right(W2, x) & right(L2, x) -> eaten.
 left(W3, x) & left(L2, x) & left(L3, x) & right(W1, x) & right(W2, x) & right(L1, x) -> eaten.

 % one wildebeest and three lions on the same bank
 right(W1, x) & right(L1, x) & right(L2, x) & right(L3, x) & left(W2, x) & left(W3, x) -> eaten.
 left(W1, x) & left(L1, x) & left(L2, x) & left(L3, x) & right(W2, x) & right(W3, x) -> eaten.

```

```
38   right(W2, x) & right(L1, x) & right(L2, x) & right(L3, x) & left(W1, x) & left(W3, x) -> eaten.
39   left(W2, x) & left(L1, x) & left(L2, x) & left(L3, x) & right(W1, x) & right(W3, x) -> eaten.
40
41   right(W3, x) & right(L1, x) & right(L2, x) & right(L3, x) & left(W1, x) & left(W2, x) -> eaten.
42   left(W3, x) & left(L1, x) & left(L2, x) & left(L3, x) & right(W1, x) & right(W2, x) -> eaten.
43
44   % two wildebeest and three lions on the same bank
45   right(W1, x) & right(W2, x) & right(L1, x) & right(L2, x) & right(L3, x) & left(W3, x) -> eaten.
46   right(W1, x) & right(W3, x) & right(L1, x) & right(L2, x) & right(L3, x) & left(W2, x) -> eaten.
47   right(W2, x) & right(W3, x) & right(L1, x) & right(L2, x) & right(L3, x) & left(W1, x)-> eaten.
48
49   left(W1, x) & left(W2, x) & left(L1, x) & left(L2, x) & left(L3, x) & right(W3, x) -> eaten.
50   left(W1, x) & left(W3, x) & left(L1, x) & left(L2, x) & left(L3, x) & right(W2, x) -> eaten.
51   left(W2, x) & left(W3, x) & left(L1, x) & left(L2, x) & left(L3, x) & right(W1, x) -> eaten.
52
53
54   left(x, y) <-> -right(x, y).
55   right(x, y) <-> -left(x, y).
56
57   left(L1, 1) & left(L2, 1) & right(L3, 1) & right(W1, 1) & right(W2, 1) & right(W3, 1).
58   left(L2, 2) & right(L1, 2) & right(L3, 2) & right(W1, 2) & right(W2, 2) & right(W3, 2).
59   left(L1, 3) & left(L2, 3) & left(L3, 3) & right(W1, 3) & right(W2, 3) & right(W3, 3).
60   left(L2, 4) & left(L3, 4) & right(L1, 4) & right(W1, 4) & right(W2, 4) & right(W3, 4).
61   left(L2, 5) & left(L3, 5) & left(W1, 5) & left(W2, 5) & right(L1, 5) & right(W3, 5).
62   left(L3, 6) & left(W2, 6) & right (L2, 6) & right(W1, 6) & right(L1, 6) & right(W3, 6).
63   left(L3, 7) & left(W1, 7) & left(W2, 7) & left(W3, 7) & right(L1, 7) & right(L2, 7).
64   left(W1, 8) & left(W2, 8) & left(W3, 8) & right(L3, 8) & right(L1, 8) & right(L2, 8).
65   left(W1, 9) & left(W2, 9) & left(W3, 9) & left(L1, 9) & left(L2, 9) & right(L3, 9).
66   left(W1, 10) & left(W2, 10) & left(W3, 10) & left(L2, 10) & right(L1, 10) & right(L3, 10).
67   left(W1, 11) & left(W2, 11) & left(W3, 11) & left(L1, 11) & left(L2, 11) & left(L3, 11).
68
69
70  end_of_list.
71
72  formulas(goals).
73   eaten.
74  end_of_list.
```

Explanation:

- To translate this problem in predicate logic, I used predicates *right(x, y)* and *left(x, y)*, which mean that animal x is on the right/left bank of the river at moment y. In lines **2 to 7**, I expressed the fact that at moment 0, all 6 animals are at the right side of the river, since they have to get to the left side. Because there are 3 lions and e wildebeest, I chose to use a constant for each of them: Li means lion number i, and Wi means wildebeest number i.
- Next, lines **9 to 36** represent all the cases in which the main rule of the problem is not met (that if the wildebeest are at any moment x outnumbered by the lions, on either side of the river, they will be eaten). These implications lead to constant *eaten* being true.
- Lines **39 and 40** translate the fact that an animal can't be on both sides of the river simultaneously, at moment y.
- The assumptions proceed in lines **42 to 52** with the process of crossing the river without leading to the wildebeest being eaten.

Commands:

- prover9 -f river_crossing.in

2.1.2 Solution Explanation and Intermediate Steps

This sub-section is dedicated to detailing the logical solution of the problem together with some additional steps that can be taken, in order to reach the final goal. Each step can be used to showcase the fact that the logic has been implemented correctly. All the intermediate goals built up the final one.

- The solution of the problem is as follows:
- There are 5 options for who goes across first: one wildebeest, one lion, two wildebeest, two lions or one of each.
- If one animal goes alone, it will just have to come straight back with the raft. And if two wildebeest cross first, the remaining one will immediately get eaten, so those options are all out.
  In order to demonstrate these dead ends, I took as an example sending two wildebeest first, which means replacing line 42 with this line : *left(W1, 1) & left(W2, 1) & right(W3, 1) & right(L1, 1) & right(L2, 1) & right(L3, 1).* and prover9 could easily prove that *eaten* will be true.
- Next, sending two lions or one of each animal actually both lead to solutions in the same number of moves, so I'm going to focus on the first one.
- Two lions cross the river at moment 1, then one of them has to come back with the raft at moment 2.
- Now, if one lion and one wildebeest cross, that one wildebeest will be outnumbered by the lions on the left side. So the only option is to send two lions on the left side at moment 3.
- One of the lions on the left side must come back with the raft at moment 4.
- As we've previously seen, we can't send one lion and one wildebeest, so the only option is to send two wildebeest on the left bank at moment 5.
- At moment 6, we can't send one wildebeest back, because the remaining one will be outnumbered. We also can't send one lion, because then the wildebeest on the right side will be outnumbered. Therefore, the only suitable step is to send one lion and one wildebeest.
- Step 7 is to get all the wildebeest on the left side.
- At moments 8, 9, 10 and 11 all we have to do is get the remaining two lions from the left side, by sending the lion on the left side step by step.
- As we run the command for prover9, we can see that with this solution, *eaten* can't be proven to be true, which is exactly what we wanted.

2.1.3 Personal observations and notes

I chose to represent this problem in predicate logic, because predicates turned out to be useful when I expressed crossing the river from side to side dependant on the moment of the solution. Also, prover9 is the most useful in this regard, as I tried to give a demonstration and prove that is a correct one.

3 Problem 3

3.1 Crystals Puzzle

In this section the solution for the following problem will be presented:

Four thieves broke into the Element Temple to steal the four magical crystals, but when the alarm went off, they panicked and each of them swallowed a crystal.

After being caught, with your secret skills of deduction **try to figure out who swallowed which crystal**.

The Water and Earth crystals forces the person to tell the truth, while the Air and Fire crystal forced the person to lie.

1. Sumi says Rikku ate the Water crystal.
2. Rikku says that Bella ate the Fire crystal.
3. Bella says that Jonah ate the Air crystal.
4. Jonah says that Sumi did not eat the Earth crystal.

3.1.1 Code implementation

Code:

```
assign(domain_size, 4).

list(distinct).
   [Sumi,Rikku,Bella,Jonah].
   [Water,Earth,Air,Fire].
end_of_list.

formulas(assumptions).

    -(Sumi = Water).
    Sumi = Earth -> Rikku = Water.
    Sumi = Air -> -(Rikku = Water).
    Sumi = Fire -> -(Rikku = Water).

    Rikku = Water -> Bella = Fire.
    Rikku = Earth -> Bella = Fire.
    Rikku = Air -> -(Bella = Fire).
    Rikku = Fire -> -(Bella = Fire).

    Bella = Water -> Jonah = Air.
    Bella = Earth -> Jonah = Air.
    Bella = Air -> -(Jonah = Air).
    Bella = Fire -> -(Jonah = Air).

    Jonah = Water -> -(Sumi = Earth).
    Jonah = Earth -> -(Sumi = Earth).
    Jonah = Air -> Sumi = Earth.
    Jonah = Fire -> Sumi = Earth.

end_of_list.
```

Explanation:

- The implementation I chose is:

- First, I decided I need 8 different constants for solving this problem. The names of the thieves can be easily mapped to the names of the crystals, since each of them has eaten no more that one crystal.
- Heading into the clues, I continued to exhaust each possibility of a thief having eaten a certain crystal.
- In lines **10 to 13**, Sumi's possibilities are presented. The equality between x and y translates into *X ate Y crystal*.
- If Sumi ate the Earth crystal, that means she tells the truth and Rikku ate the Water crystal. If she ate the Air or the Fire crystal it means she's lying, so Rikku didn't eat the Water crystal.
- Solving this puzzle myself, I realized that line 10 must be added, because if Sumi ate the Water crystal implies that her statement has to be true. At the same time, we know that each of thief ate a different crystal so Sumi and Rikku couldn't have eaten the same crystal.
- Next, I added the possibilities for Rikku. If Rikku ate the Water or the Earth crystal, his statement is true so Bella ate the Fire crystal. On the other hand, if he ate the Air or the Fire crystal, Bella didn't eat the Fire one.
- For Bella, we have the following ways: if she ate the Water or the Earth crystal, Jonah ate the Air one, and if she ate the Air or the Fire one, Jonah didn't eat the Air crystal.
- When it comes to Jonah, we can say that if he ate the Water or the Earth crystal, Sumi didn't eat the Earth one, and if he ate the Air or Fire crystal, Sumi did eat the Earth crystal.

Commands:

- mace4 -m -1 -f crystals.in

3.1.2 Solution Explanation and Intermediate Steps

This sub-section is dedicated to detailing the logical solution of the problem together with some additional steps that can be taken, in order to reach the final goal. Each step can be used to showcase the fact that the logic has been implemented correctly. All the intermediate goals built up the final one.

- In order to check the validity of the model given by mace4, we have to solve the puzzle, to find the possible solutions.
- Trying to prove a contradiction will work the best in this case.
- For example, considering that Sumi is telling the truth, Rikku would have the Water crystal, and since she's telling the truth she would have Earth, so according to Rikku, Bella would have the Fire crystal. But then Bella would have to be lying about Jonah having the air crystal, and yet that's the only one remaining option. Therefore our initial assumption is wrong.
- Now we know that Sumi, as a liar, would have the Fire or Air crystal, that means Jonah was telling the truth about her, so he can't have taken either of those, and that means Bella was lying about him, so she must also have the Fire or Air.
- Since Sumi was lying, Rikku can't have taken the Water crystal, and the only one left who could have it is Jonah.
- Because we already identified the two liars (Sumi and Bella), Rikku must have the Earth crystal. That means he told the truth about Bella having the Fire crystal.
- In the end, we are left with Sumi having the Air crystal.

3.1.3 Personal observations and notes

This problem is solved perfectly by trial and error, and this is the reason why I chose to use mace4 for this puzzle, that results in one simple solution. The answer I've given above is exactly the model that mace gives.

4 Problem 4

4.1 Troll Paradox

In this section the solution for the following problem will be presented:

You and your brother are exploring the wonderful world of Paradoxica. Fantastically paradoxical creatures fly around you.

Suddenly, a troll catches all the creatures in an enormous net. You bravely step forward and demand it let them go, when the troll grabs your brother and puts him in a cage.

Then he makes you an offer: if you say something false, then he'll release your brother, if you say something true, he'll release the creatures. The troll hates paradoxes, so don't try to cheat and say something paradoxical.

What true/false statement can you say to force the troll to free your brother and the paradoxical creatures?

We know:

1. If you say a true statement, the troll will release the creatures.
2. If you say a false statement, the troll will release your brother.
3. Saying something that is neither true nor false won't work because the troll hates paradoxes.

4.1.1 Code implementation

Code:

```
1  formulas(assumptions).
2
3   % define "true" and "false" statement
4   true(x) -> -false(x).
5   false(x) -> -true(x).
6
7   %if the troll considers releasing person x to be true, he must release them
8   %if the troll considers releasing person x to be false, he must not release them
9   %in order to not create a paradox
10  true(x) -> release(x).
11  false(x) -> -release(x).
12
13  true(x) -> release(Creatures).
14  false(x) -> release(Brother).
15
16  true(Brother). % the troll hates paradoxes, so he must consider releasing your brother to be true
17
18 end_of_list.
19
20 formulas(goals).
21  release(Creatures) & release(Brother).
22 end_of_list.
```

Explanation:

- The predicate logic translation of this paradox is as follows:
- In lines **4 and 5** I defined the predicates *true and false* that I used to express what the troll considers the statement to be.
- In lines **10 and 11** are the implications that have to be respected in order for the statement to not lead to a paradox, which we don't want, according to the clues: if the troll considers releasing person x to be true, he must release them and also if it considers releasing the person x to be false, he must not release them. Not meeting these implications results in a paradox.

- The constants of this problem are Creatures and Brother, that we need to express the statement. Next, in lines **13 and 14** I've completed the troll's rules which say that if the statement is true, it will release the Creatures, respectively if it's false, it will release the Brother.
- Line **16** contains the statement that we need to tell the troll in order to save the creatures and the brother. *true(Brother)* translates in: the troll considers you telling him *"You will release my brother"* to be *true*.
- The goal we want to reach and demonstrate that is obtainable is releasing both the creatures and the brother, which is presented in line **21**.

Commands:

- pover9 -f troll_paradox.in

4.1.2 Solution Explanation and Intermediate Steps

This sub-section is dedicated to detailing the logical solution of the problem together with some additional steps that can be taken, in order to reach the final goal. Each step can be used to showcase the fact that the logic has been implemented correctly. All the intermediate goals built up the final one.

- Because we want the troll to free both the creatures and the brother, we must choose who we want him to release.
- If we replace the assumption from line 16 with *true(Creatures)*, that means that we told the troll that he would release the creatures. He considers it to be true, so he'll release the creatures but not the brother and the goal *release(Creatures)* will be proven but not *release(Brother)*.
- If we replace it with *false(Creatures)*, then he considers letting go of the creatures to be false and he will release only the brother, therefore *release(Brother)* will be proven.
- In the end, we cannot replace it with *false(Brother)*, because that will lead to a paradox: if the troll considers that "You will release my brother" is false, then by it's own rules, he'll have to release the brother. The only suitable assumption is for the troll to consider releasing the brother to be true, which by the rules means that he'll release the creatures too.

4.1.3 Personal observations and notes

Because I approached this paradox with a demonstration technique, the best option was to use prover9, to help me reach the goal.