Artificial intelligence - Project 2
- Propositional logic -

Ilovan Bianca-Maria

8/12/2020

1 Logic and Zebra Puzzles

1.1 Problem 1 - Dinner Date

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

Every anniversary, 4 couples return to the spot they had their first date and order the same meal each had. As a further token of celebration, each husband brings along a bouquet of flowers that match the ones he originally brought to their first date. From the clues provided, determine the date of the anniversary, the first name of each woman, the drink she ordered and the type of flower bouquet she was given.

We have the following clues, that will help us to solve the problem:
1.The 4 people were Norma, the person whose favorite flowers are forget-me-nots, the person whose anniversary is on January 17th, and the one who drinks vodka.
2.The person whose favorite flowers are Purslanes is not April.
3.The person whose anniversary is on April 24th love carnations.
4.Either the person whose anniversary is on April 24th or the person whose anniversary is on September 3rd,loves to drink champagne.
5.Of April and Norma, one has an anniversary on April 24th and the other enjoys drinking gin.
6.The woman who ordered champagne has an anniversary after Melissa.
7.The person whose favorite flowers are forget-me-nots is not Bethany.
8.Bethany has an anniversary after Melissa.
9.The woman who received forget-me-nots enjoys drinking rum.

1.1.1 Code implementation

This sub-section is dedicated to showcasing your own solution that you came up with for solving the above question. One has to put here any **code** that has been used for solving the above task, along with **comments** that explain every design decision made. To reference the code, please make use of the *code lines number.*

Code:

```
% Saved by Prover9-Mace4 Version 0.5, December 2007.

set(ignore_option_dependencies). % GUI handles dependencies

if(Prover9). % Options for Prover9
  assign(max_seconds, 60).
end_if.

if(Mace4).   % Options for Mace4
  assign(max_seconds, 60).
end_if.

formulas(assumptions).

%Every anniversary, 4 couples return to the spot they had their first date and order the same meal each
%As a further token of celebration, each husband brings along a bouquet of flowers that match the ones h
%From the clues provided, determine:
% - the date of the anniversary
% - the first name of each woman
% - the drink she ordered
% - the type of flower bouquet she was given.

%differrentFrom(x,y)
```

```
%girl1,girl2,girl3,girl4

differentFrom(girl1,girl2).
differentFrom(girl1,girl3).
differentFrom(girl1,girl4).
differentFrom(girl2,girl3).
differentFrom(girl2,girl4).
differentFrom(girl3,girl4).

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

%after(x,y)
after(girl1,girl2).
after(girl1,girl3).
after(girl1,girl4).
after(girl2,girl3).
after(girl2,girl4).
after(girl3,girl4).

%notAfter(x,y)
-after(girl1,girl1).
-after(girl2,girl1).
-after(girl2,girl2).
-after(girl3,girl1).
-after(girl3,girl2).
-after(girl3,girl3).
-after(girl4,girl1).
-after(girl4,girl2).
-after(girl4,girl3).
-after(girl4,girl4).

%Girls
April(x) | Bethany(x) | Melissa(x) | Norma(x).
%Flowers
Carnations(x) | ForgetMeNot(x) | LeopardsBan(x) | Purslane(x).
%Drinks
Champagne(x) | Gin(x) | Rum(x) | Vodka(x).
%Anniversary
Jan17(x) | April14(x) | April24(x) | Sept3(x).

%girls
April(x) & April(y) -> -differentFrom(x,y).
Bethany(x) & Bethany(y) -> -differentFrom(x,y).
Melissa(x) & Melissa(y) -> -differentFrom(x,y).
Norma(x) & Norma(y) -> -differentFrom(x,y).

%flowers
Carnations(x) & Carnations(y) -> -differentFrom(x,y).
ForgetMeNot(x) & ForgetMeNot(y) -> -differentFrom(x,y).
LeopardsBan(x) & LeopardsBan(y) -> -differentFrom(x,y).
Purslane(x) & Purslane(y) -> -differentFrom(x,y).

%drinks
Champagne(x) & Champagne(y) -> -differentFrom(x,y).
```

```
78  Gin(x) & Gin(y) -> -differentFrom(x,y).
79  Rum(x) & Rum(y) -> -differentFrom(x,y).
80  Vodka(x) & Vodka(y) -> -differentFrom(x,y).
81
82  %anniversary
83  Jan17(x) & Jan17(y) -> -differentFrom(x,y).
84  April14(x) & April14(y) -> -differentFrom(x,y).
85  April24(x) & April24(y) -> -differentFrom(x,y).
86  Sept3(x) & Sept3(y) -> -differentFrom(x,y).
87
88  % Clues.
89
90  %1.The 4 people were Norma, the person whose favorite flowers are forget-me-nots, the person whose anniv
91  Norma(x) | ForgetMeNot(x) | Jan17(x) | Vodka(x).
92
93  %2.The person whose favorite flowers are Purslanes is not April.
94  Purslane(x) -> -April(x).
95
96  %3.The person whose anniversary is on April 24th love carnations.
97  April24(x) -> Carnations(x).
98
99  %4.Either the person whose anniversary is on April 24th or the person whose anniversary is on September
100 Champagne(x) -> April24(x) | Sept3(x).
101
102 %5.Of April and Norma,one has an anniversary on April 24th and the other enjoys drinking gin.
103 April(x) & Norma(y) -> (April24(x) & Gin(y)) | (April24(y) & Gin(x)).
104
105 %6.The woman who ordered champagne has an anniversary after Melissa.
106 Champagne(x) & Melissa(y) -> after(y,x).
107
108 %7.The person  whose favorite flowers are forget-me-nots is not Bethany.
109 ForgetMeNot(x) -> -Bethany(x).
110
111 %8.Bethany has an anniversary after Melissa.
112 Bethany(x) & Melissa(y) -> after(y,x).
113
114 %9.The woman who received forget-me-nots enjoys drinking rum.
115 ForgetMeNot(x) -> Rum(x).
116
117 end_of_list.
118
119 formulas(goals).
120
121 end_of_list.
```

Explanation:

- in lines 26-31, using the predicate differentFrom(x,y), we want to express that our 4 girls are different from one another.
- in line 33, we showed that the "differentFrom" relation is a symmetrical one.
- in the next lines, we use the predicate after(x,y), meaning that x is somewhere to the left of y.
- the relation -after(x,x) means that girl1 cannot be the left neighbor of girl1, and in the lines 44-53 we treat the rest of our cases for the 4 girls.

- immediately after that, we've expressed that we have a name, a flower, a drink and an anniversary for each of our 4 girls. Each girl is unique (lines 64-86).
- in the next lines, with the given clues, we tried to treat separately each clue.

Results:

```
interpretation( 4, [number = 1,seconds = 0], [
    function(girl1, [0]),
    function(girl2, [1]),
    function(girl3, [2]),
    function(girl4, [3]),
    relation(April(_), [0,0,0,1]),
    relation(April14(_), [0,0,1,0]),
    relation(April24(_), [0,1,0,0]),
    relation(Bethany(_), [0,0,1,0]),
    relation(Carnations(_), [0,1,0,0]),
    relation(Champagne(_), [0,1,0,0]),
    relation(ForgetMeNot(_), [1,0,0,0]),
    relation(Gin(_), [0,0,0,1]),
    relation(Jan17(_), [0,0,0,1]),
    relation(LeopardsBan(_), [0,0,0,1]),
    relation(Melissa(_), [1,0,0,0]),
    relation(Norma(_), [0,1,0,0]),
    relation(Purslane(_), [0,0,1,0]),
    relation(Rum(_), [1,0,0,0]),
    relation(Sept3(_), [1,0,0,0]),
    relation(Vodka(_), [0,0,1,0]),
    relation(after(_,_), [
        0,1,1,1,
        0,0,1,1,
        0,0,0,1,
        0,0,0,0]),
    relation(differentFrom(_,_), [
        0,1,1,1,
        1,0,1,1,
        1,1,0,1,
        1,1,1,0])]).
```

Conclusion:

April - Jan 17 th - Leopard's Ban - Gin.
Melissa - April 14 th -> Forget-me-not -> Rum.
Norma -> April 24 th -> Carnations -> Champagne.
Bethany -> Sept 3 rd -> Purslane -> Vodka.

1.2 Problem 2 - Travel Agency

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

Five women are side by side booking a trip in a travel agency. Each one is traveling to a different country. Follow the clues to discover where each one is going.

We have the following clues, that will help us to solve the problem:
1. The singer is at the third position.
2.Woman traveling for 20 days is somewhere between the woman who is going to Peru and the owner of the Blue purse.
3.Ana is exactly to the left of the Biologist.
4.The 32 year-old is going to see the Sahara.
5.The owner of the white purse is exactly to the right of the woman traveling to Machu Picchu(Peru).
6.Glenda is to the right of the woman who has green purse.
7.The woman with the white purse is somewhere between 30 years old woman and the owner of the Blue purse, in this order.
8.The 24 year-old woman is going to visit Aztec pyramid(Mexico).
9.The woman wearing White purse is somewhere to the left of the youngest woman.
10.The traveler going to Italy is exactly to the right of the woman traveling for 20 days.
11.The person who is going to travel for 25 days has the red purse.
12. The judge is in the first position.
13.The nurse is exactly to the right of the woman who is going to travel for 20 days.
14.Hostess-between Lara and the woman with blue purse.
15.In the second position is the woman that is going to travel for 15 days.
16. Rose has the green purse.
17. Less than a week, to the left of the 32 year-old woman.
18. The person traveling for 5 days is 28.
19.Blue purse owner-between 30 year old woman and the owner of yellow purse

1.2.1 Code implementation

This sub-section is dedicated to showcasing your own solution that you came up with for solving the above question. One has to put here any **code** that has been used for solving the above task, along with **comments** that explain every design decision made. To reference the code, please make use of the *code lines number*. Additionally, complete this sub-section with any **command configurations** that you may have used during the implementation or testing process (please fill in *just the arguments*).

Code:

```
% Saved by Prover9-Mace4 Version 0.5, December 2007.

set(ignore_option_dependencies). % GUI handles dependencies

if(Prover9). % Options for Prover9
  assign(max_seconds, 60).
end_if.

if(Mace4).   % Options for Mace4
  assign(max_seconds, 60).
end_if.

formulas(assumptions).

%Five women are side by side booking a trip in a travel agency.
%Each one is traveling to a different country.
%Follow the clues to discover where each one is going.

%differrentFrom(x,y)
%woman1,woman2,woman3,woman4, woman5

```

```
differentFrom(Girl1,Girl2).
differentFrom(Girl1,Girl3).
differentFrom(Girl1,Girl4).
differentFrom(Girl1,Girl5).
differentFrom(Girl2,Girl3).
differentFrom(Girl2,Girl4).
differentFrom(Girl2,Girl5).
differentFrom(Girl3,Girl4).
differentFrom(Girl3,Girl5).
differentFrom(Girl4,Girl5).

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

%leftneighbor(x,y), x e exact in stanga lui y.
leftneighbor(Girl1,Girl2).
leftneighbor(Girl2,Girl3).
leftneighbor(Girl3,Girl4).
leftneighbor(Girl4,Girl5).

%notFirstLeftNeighbor
-leftneighbor(Girl1,Girl1).
-leftneighbor(Girl2,Girl1).
-leftneighbor(Girl2,Girl2).
-leftneighbor(Girl3,Girl1).
-leftneighbor(Girl3,Girl2).
-leftneighbor(Girl3,Girl3).
-leftneighbor(Girl4,Girl1).
-leftneighbor(Girl4,Girl2).
-leftneighbor(Girl4,Girl3).
-leftneighbor(Girl4,Girl4).
-leftneighbor(Girl4,Girl4).
-leftneighbor(Girl5,Girl1).
-leftneighbor(Girl5,Girl2).
-leftneighbor(Girl5,Girl3).
-leftneighbor(Girl5,Girl4).
-leftneighbor(Girl5,Girl5).

%after(x,y), x e undeva in stanga lui y.
after(Girl1,Girl2).
after(Girl1,Girl3).
after(Girl1,Girl4).
after(Girl1,Girl5).
after(Girl2,Girl3).
after(Girl2,Girl4).
after(Girl2,Girl5).
after(Girl3,Girl4).
after(Girl3,Girl5).
after(Girl4,Girl5).

%notFirstAfter(x,y)
-after(Girl1,Girl1).
-after(Girl2,Girl1).
-after(Girl2,Girl2).
-after(Girl3,Girl1).
```

```
-after(Girl3,Girl2).
-after(Girl3,Girl3).
-after(Girl4,Girl1).
-after(Girl4,Girl2).
-after(Girl4,Girl3).
-after(Girl4,Girl4).
-after(Girl4,Girl4).
-after(Girl5,Girl1).
-after(Girl5,Girl2).
-after(Girl5,Girl3).
-after(Girl5,Girl4).
-after(Girl5,Girl5).

%between(x,y,z)
between(x,y,z) -> after(x,y) & after(y,z) & after(x,z). %y e in dreapta lui x, y e in stanga lui z si z
between2(x,y,z) -> leftneighbor(x,y) & leftneighbor(y,z) & after(x,z).

%Names
Ana(x) | Glenda(x) | Jessie(x) | Lara(x) | Rose(x).

%Age
Age24(x) | Age26(x) | Age28(x) | Age30(x) | Age32(x).

%Profession
Biologist(x) | Hostess(x) | Judge(x) | Nurse(x) | Singer(x).

%Country
China(x) | Egipt(x) | Italy(x) | Mexico(x) | Peru(x).

%Duration
Days5(x) | Days10(x) | Days15(x) | Days20(x) | Days25(x).

%Purse
Blue(x) | Green(x) | Red(x) | White(x) | Yellow(x).

%women
Ana(x) & Ana(y) -> -differentFrom(x,y).
Glenda(x) & Glenda(y) -> -differentFrom(x,y).
Jessie(x) & Jessie(y) -> -differentFrom(x,y).
Lara(x) & Lara(y) -> -differentFrom(x,y).
Rose(x) & Rose(y) -> -differentFrom(x,y).

%age
Age24(x) & Age24(y) -> -differentFrom(x,y).
Age26(x) & Age26(y) -> -differentFrom(x,y).
Age28(x) & Age28(y) -> -differentFrom(x,y).
Age30(x) & Age30(y) -> -differentFrom(x,y).
Age32(x) & Age32(y) -> -differentFrom(x,y).

%proffesions
Biologist(x) & Biologist(y) -> -differentFrom(x,y).
Hostess(x) & Hostess(y) -> -differentFrom(x,y).
Judge(x) & Judge(y) -> -differentFrom(x,y).
Nurse(x) & Nurse(y) -> -differentFrom(x,y).
```

```
Singer(x) & Singer(y) -> -differentFrom(x,y).

%countries
China(x) & China(y) -> -differentFrom(x,y).
Egipt(x) & Egipt(y) -> -differentFrom(x,y).
Italy(x) & Italy(y) -> -differentFrom(x,y).
Mexico(x) & Mexico(y) -> -differentFrom(x,y).
Peru(x) & Peru(y) -> -differentFrom(x,y).

%duration
Days5(x) & Days5(y) -> -differentFrom(x,y).
Days10(x) & Days10(y) -> -differentFrom(x,y).
Days15(x) & Days15(y) -> -differentFrom(x,y).
Days20(x) & Days20(y) -> -differentFrom(x,y).
Days25(x) & Days25(y) -> -differentFrom(x,y).

Blue(x) & Blue(y) -> -differentFrom(x,y).
Green(x) & Green(y) -> -differentFrom(x,y).
Red(x) & Red(y) -> -differentFrom(x,y).
White(x) & White(y) -> -differentFrom(x,y).
Yellow(x) & Yellow(y) -> -differentFrom(x,y).

%Clues
%1. The singer is at the third position.
Singer(Girl3).

%2.Woman traveling for 20 days is somewhere between the woman who is going to Peru
%and the owner of the Blue purse.
Peru(x) & Days20(y) & Blue(z)-> between(x,y,z).

%3.Ana is exactly to the left of the Biologist.
Ana(x) & Biologist(y) -> leftneighbor(x,y).

%4.The 32 year-old is going to see the Sahara.
Age32(x) <-> Egipt(x).

%5.The owner of the white purse is exactly to the right of the woman traveling to Machu Pichu(Peru)
White(x) & Peru(y) -> leftneighbor(y,x).

%6.Glenda is to the right of the woman who has green purse
Glenda(x) & Green(y) -> after(y,x).

%7.White purse->between 30 years old woman and the owner of the Blue purse.
Age30(x) & White(y) & Blue(z) -> between(x,y,z).

%8.The 24 year-old woman is going to visit Aztec pyramid.
Age24(x)<->Mexico(x).

%9.The woman wearing White purse is somewhere to the left of the yougest woman
White(x) & Age24(y) -> after(x,y).

%10.The traveler going to Italy is exactly to the right of the woman traveling for 20 days.
Italy(x) & Days20(y) -> leftneighbor(y,x).

```

```
184 %11.The person who is going to travel for 25 days has the red purse.
185 Days25(x) <-> Red(x).
186
187 %12. The judge is in the first position.
188 Judge(Girl1).
189
190 %13.The nurse is exactly to the right of the woman who is going to travel for 20 days.
191 Nurse(x) & Days20(y) -> leftneighbor(y,x).
192
193 %14.Hostess-between Lara and the woman with blue purse
194 Lara(x) & Hostess(y) & Blue(z) -> between(x,y,z).
195
196 %15.In the second position is the woman that is going to travel for 15 days
197 Days15(Girl2).
198
199 %16. Rose has the green purse
200 Rose(x) <-> Green(x).
201
202 %17. Less than a week, to the left of the 32 year-old woman
203 Days5(x) & Age32(y) -> leftneighbor(x,y).
204
205 %18. The person traveling for 5 days is 28.
206 Days5(x)<->Age28(x).
207
208 %19.Blue purse owner-between 30 year old woman and the owner of yellow purse
209 Age30(x) & Blue(y) & Yellow(z) -> between(x,y,z).
210
211 end_of_list.
212
213 formulas(goals).
214
215 end_of_list.
```

Explanation:

- in lines 22-31, using the predicate differentFrom(x,y), we want to express that our 5 women are different from one another. Our 5 women are: Girl1, Girl2, Girl3, Girl4, Girl5.
- in line 33, we showed that the "differentFrom" relation is a symmetrical one.
- in the next lines, we use the predicate leftneighbor(x,y), meaning that x is exactly to the left of y.
- the relation -leftneighbor(x,x) means that Girl1 cannot be the left neighbor of Girl1, and in the lines 24-39 we treat the rest of our cases.
- in lines 60-69, we use the predicate after(x,y), meaning that x is somewhere to the left of y.
- predicate -after(x,x) means that Girl1 cannot be the left neighbor of Girl1, and in the lines 72-87 we treat the rest of our cases.
- in line 90, we define the predicate between(x,y,z), meaning that y should be between x and z. With after(x,y),after(y,z), after(x,z), we can say that x is somewhere to the left of y, y somewhere to the left of z, and x is somewhere to the left of z.
- immediately after that, we've expressed that we have a name, an age, a profession, a country, a duration and a color of the purse for each of our 5 women. Each girl is unique (lines 111-150).
- in the next lines, with the given clues, we tried to treat separately each clue.

Results:

```
interpretation( 5, [number = 1,seconds = 0], [
    function(Girl1, [0]),
    function(Girl2, [1]),
    function(Girl3, [2]),
    function(Girl4, [3]),
    function(Girl5, [4]),
    relation(Age24(_), [0,0,1,0,0]),
    relation(Age26(_), [0,1,0,0,0]),
    relation(Age28(_), [0,0,0,1,0]),
    relation(Age30(_), [1,0,0,0,0]),
    relation(Age32(_), [0,0,0,0,1]),
    relation(Ana(_), [0,0,0,1,0]),
    relation(Biologist(_), [0,0,0,0,1]),
    relation(Blue(_), [0,0,0,1,0]),
    relation(China(_), [0,1,0,0,0]),
    relation(Days10(_), [0,0,0,0,1]),
    relation(Days15(_), [0,1,0,0,0]),
    relation(Days20(_), [0,0,1,0,0]),
    relation(Days25(_), [1,0,0,0,0]),
    relation(Days5(_), [0,0,0,1,0]),
    relation(Egipt(_), [0,0,0,0,1]),
    relation(Glenda(_), [0,0,0,0,1]),
    relation(Green(_), [0,0,1,0,0]),
    relation(Hostess(_), [0,1,0,0,0]),
    relation(Italy(_), [0,0,0,1,0]),
    relation(Jessie(_), [0,1,0,0,0]),
    relation(Judge(_), [1,0,0,0,0]),
    relation(Lara(_), [1,0,0,0,0]),
    relation(Mexico(_), [0,0,1,0,0]),
    relation(Nurse(_), [0,0,0,1,0]),
    relation(Peru(_), [1,0,0,0,0]),
    relation(Red(_), [1,0,0,0,0]),
    relation(Rose(_), [0,0,1,0,0]),
    relation(Singer(_), [0,0,1,0,0]),
    relation(White(_), [0,1,0,0,0]),
    relation(Yellow(_), [0,0,0,0,1]),
    relation(after(_,_), [
        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]),
    relation(differentFrom(_,_), [
        0,1,1,1,1,
        1,0,1,1,1,
        1,1,0,1,1,
        1,1,1,0,1,
        1,1,1,1,0]),
    relation(leftneighbor(_,_), [
        0,1,0,0,0,
        0,0,1,0,0,
        0,0,0,1,0,
        0,0,0,0,1,
        0,0,0,0,0]),
    relation(between(_,_,_), [0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    relation(between2(_,_,_), [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Conclusion:

Woman 1: Lara -> Judge -> 30 years -> Peru -> 25 days -> red.
Woman 2: Jessie -> Hostess -> 26 years -> China -> 15 days -> white.
Woman 3: Rose -> Singer -> 24 years -> Mexico -> 20 days -> green.
Woman 4: Ana -> Nurse -> 28 years -> Italy -> 5 days -> blue.
Woman 5: Glenda -> Biologist -> 32 years -> Egypt -> 10 days -> yellow.

1.3 Problem 3 - Home Sick

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

Five students in Mrs. Hill's class each became ill on a certain day of the week and had to stay at home for a number of days. While the children were home sick, their mothers made them each their favorite soup for lunch each day while they watched cartoons. From the clues provided, determine which child came down with which illness, which soup they were served and on which day he or she did not go to school.

We have the following clues that will help us to solve the problem:
1.The person who had the fever is not Molly.
2.Nancy did not have fever.
3.The person who had chicken pox left school sometime before Alex
4.The child who did not return to school on Monday did not have migraine.
5.Jacob was unfortunate to come down with a nasty case of the flu.
6.The five students are Molly,the person who had a migraine, the student who has served beef and barley soup,the person who left school on Thursday and Alex.
7.The child who first missed school on Thursday wasn't served minestrone soup.
8.The child who had the fever was not Alex.
9.Molly was given a streaming bowl of chicken noodle soup for lunch.
10.The student who had the beef and barley soup left the day after the child whose mother fed them tomato soup.
11.The student that did not show up for class on Thursday didn't have the flu.

1.3.1 Code implementation

This sub-section is dedicated to showcasing your own solution that you came up with for solving the above question. One has to put here any **code** that has been used for solving the above task, along with **comments** that explain every design decision made. To reference the code, please make use of the *code lines number.*

Code:

```
% Saved by Prover9-Mace4 Version 0.5, December 2007.

set(ignore_option_dependencies). % GUI handles dependencies

if(Prover9). % Options for Prover9
  assign(max_seconds, 60).
```

```
end_if.

if(Mace4).   % Options for Mace4
  assign(max_seconds, 60).
end_if.

formulas(assumptions).

%Five students in Mrs. Hill's class each became ill on a certain day of the week and had to stay at home
%While the children were home sick, their mothers made them each their favorite soup for lunch each day
%From the clues provided, determine which child came down with which illness, which soup they were serve

%differentFrom(x,y)
%child1,child2,child3,child4,child5
differentFrom(child1,child2).
differentFrom(child1,child3).
differentFrom(child1,child4).
differentFrom(child1,child5).
differentFrom(child2,child3).
differentFrom(child2,child4).
differentFrom(child2,child5).
differentFrom(child3,child4).
differentFrom(child3,child5).
differentFrom(child4,child5).

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

before(child1,child2).
before(child1,child3).
before(child1,child4).
before(child1,child5).
before(child2,child3).
before(child2,child4).
before(child2,child5).
before(child3,child4).
before(child3,child5).
before(child4,child5).

-before(child1,child1).
-before(child2,child1).
-before(child2,child2).
-before(child3,child1).
-before(child3,child2).
-before(child3,child3).
-before(child4,child1).
-before(child4,child2).
-before(child4,child3).
-before(child4,child4).
-before(child5,child1).
-before(child5,child2).
-before(child5,child3).
-before(child5,child4).
-before(child5,child5).

```

```
Alex(x) | Betty(x) | Jacob(x) | Molly(x) | Nancy(x).
Monday(x) | Tuesday(x) | Wednesday(x) | Thursday(x) | Friday(x).
ChickenPox(x) | Cold(x) | Fever(x) | Flu(x) | Migraine(x).
BeanAndBacon(x) | BeefAndBarley(x) | ChickenNoodle(x) | Minestrone(x) | Tomato(x).

Alex(x) & Alex(y) -> -differentFrom(x,y).
Betty(x) & Betty(y) -> -differentFrom(x,y).
Jacob(x) & Jacob(y) -> -differentFrom(x,y).
Molly(x) & Molly(y) -> -differentFrom(x,y).
Nancy(x) & Nancy(y) -> -differentFrom(x,y).

Monday(x) & Monday(y) -> -differentFrom(x,y).
Tuesday(x) & Tuesday(y) -> -differentFrom(x,y).
Wednesday(x) & Wednesday(y) -> -differentFrom(x,y).
Thursday(x) & Thursday(y) -> -differentFrom(x,y).
Friday(x) & Friday(y) -> -differentFrom(x,y).

ChickenPox(x) & ChickenPox(y) -> -differentFrom(x,y).
Cold(x) & Cold(y) -> -differentFrom(x,y).
Fever(x) & Fever(y) -> -differentFrom(x,y).
Flu(x) & Flu(y) -> -differentFrom(x,y).
Migraine(x) & Migraine(y) -> -differentFrom(x,y).

BeanAndBacon(x) & BeanAndBacon(y) -> -differentFrom(x,y).
BeefAndBarley(x) & BeefAndBarley(y) -> -differentFrom(x,y).
ChickenNoodle(x) & ChickenNoodle(y) -> -differentFrom(x,y).
Minestrone(x) & Minestrone(y) -> -differentFrom(x,y).
Tomato(x) & Tomato(y) -> -differentFrom(x,y).

%Clues
%1.The person who had the fever is not Molly.
Fever(x) -> -Molly(x).

%2.Nancy did not have fever.
Nancy(x) -> -Fever(x).

%3.The person who had chicken pox left school sometime before Alex
Monday(x) & Tuesday(y) -> before(x,y).
Monday(x) & Wednesday(y) -> before(x,y).
Monday(x) & Thursday(y) -> before(x,y).
Monday(x) & Friday(y) -> before(x,y).
Tuesday(x) & Wednesday(y) -> before(x,y).
Tuesday(x) & Thursday(y) -> before(x,y).
Tuesday(x) & Friday(y) -> before(x,y).
Wednesday(x) & Thursday(y) -> before(x,y).
Wednesday(x) & Friday(y) -> before(x,y).
Thursday(x) & Friday(y) -> before(x,y).

ChickenPox(x) & Alex(y) -> before(x,y).

%4.The child who did not return to school on Monday did not have migraine.
Monday(x) -> -Migraine(x).

%5.Jacob was unfortunate to come down with a nasty case of the flu.
```

```
115 Jacob(x) -> Flu(x).
116
117 %6.The five students are Molly, the person who had a migraine, the student who has served beef and barle
118 Molly(x) | Migraine(x) | BeefAndBarley(x) | Thursday(x) | Alex(x).
119
120 %7.The child who first missed school on Thursday wasn't served minestrone soup.
121 Thursday(x) -> -Minestrone(x).
122
123 %8.The child who had the fever was not Alex.
124 Fever(x) -> -Alex(x).
125
126 %9.Molly was given a streaming bowl of chicken noodle soup for lunch.
127 Molly(x) -> ChickenNoodle(x).
128
129 %10.The student who had the beef and barley soup left the day after the child whose mother fed them toma
130 BeefAndBarley(x) & Tomato(y) -> before(y,x).
131
132 %11.The student that did not show up for class on Thursday didn't have the flu.
133 Thursday(x) -> -Flu(x).
134
135 end_of_list.
136
137 formulas(goals).
138
139 end_of_list.
```

Explanation:

- in lines 21-30, using the predicate differentFrom(x,y), we want to express that our 5 children are different from one another. Our 5 children are: child1, child2, child3, child4, child5.
- in line 32, we showed that the "differentFrom" relation is a symmetrical one.
- in the next lines, we use the predicate before(x,y), meaning that x is somewhere to the left of y.
- the relation -before(x,x) means that child1 cannot be the left neighbor of child1, and in the lines 45-59 we treat the rest of our cases.
- immediately after that, we have to take into account that each child has a name and our 5 possible names are: Alex, Betty, Jacob, Molly and Nancy, a day of the week, a type of soup: Bean and bacon, beef and barley, chicken noodle, minestrone, tomato and an illness:Chicken Pox, Cold, Fever, Migraine. Each child is unique(lines 66-88).
- in the next lines are implemented the given clues.

Results:

```
interpretation( 5, [number = 1,seconds = 0], [
    function(child1, [0]),
    function(child2, [1]),
    function(child3, [2]),
    function(child4, [3]),
    function(child5, [4]),
    relation(Alex(_), [0,0,0,0,1]),
    relation(BeanAndBacon(_), [0,0,0,1,0]),
    relation(BeefAndBarley(_), [0,0,1,0,0]),
    relation(Betty(_), [0,0,0,1,0]),
    relation(ChickenNoodle(_), [1,0,0,0,0]),
    relation(ChickenPox(_), [1,0,0,0,0]),
    relation(Cold(_), [0,0,0,0,1]),
    relation(Fever(_), [0,0,0,1,0]),
    relation(Flu(_), [0,0,1,0,0]),
    relation(Friday(_), [0,0,0,0,1]),
    relation(Jacob(_), [0,0,1,0,0]),
    relation(Migraine(_), [0,1,0,0,0]),
    relation(Minestrone(_), [0,0,0,0,1]),
    relation(Molly(_), [1,0,0,0,0]),
    relation(Monday(_), [1,0,0,0,0]),
    relation(Nancy(_), [0,1,0,0,0]),
    relation(Thursday(_), [0,0,0,1,0]),
    relation(Tomato(_), [0,1,0,0,0]),
    relation(Tuesday(_), [0,1,0,0,0]),
    relation(Wednesday(_), [0,0,1,0,0]),
    relation(before(_,_), [
        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]),
    relation(differentFrom(_,_), [
        0,1,1,1,1,
        1,0,1,1,1,
        1,1,0,1,1,
        1,1,1,0,1,
        1,1,1,1,0])]).
```

Conclusion:

Molly -> Monday -> Chicken Noodle -> Chicken Pox.
Nancy -> Tuesday ->Tomato -> Migraine.
Jacob -> Wednesday -> Beef and Barley -> Flu.
Betty -> Thursday -> Bean and Bacon -> Fever.
Alex -> Friday -> Minestrone -> Cold.

1.4 Problem 4 - The disadvantages of being Absent-minded

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

It is well known that in any group of at least 23 people, the odds are greater than 50 percent that at least two of them will have the same birthday. Professor Smullyan was once teaching an undergraduate mathematics class at Princeton, discussing elementary probability theory. He explained to the class that with 30 people instead of 23, the odds would become enormously high that at least two of them had the same birthday.

"Now," the professor continued, "since there are only nineteen students in this class, the odds are much less than fifty percent that any two of you have the same birthday."

At this point one of the students raised his hand and said, "I'll bet you that at least two of us here have the same birthday."

"It wouldn't be right for me to take the bet," the professor replied, "because the probabilities would be highly in my favor."

"I don't care," said the student, "I'll bet you anyhow!"

"All right," the professor said, thinking to teach the student a good lesson. He then proceeded to call on the students one by one to announce their birthdays until, about halfway through, both the professor and the class burst out laughing at the professor's stupidity. The boy who had so confidently made the bet did not know the birthday of anyone present except his own. Can you guess why he was so confident?

Code:

```
1  % Saved by Prover9-Mace4 Version 0.5, December 2007.
2  
3  set(ignore_option_dependencies). % GUI handles dependencies
4  
5  if(Prover9). % Options for Prover9
6    assign(max_seconds, 60).
7  end_if.
8  
9  if(Mace4).   % Options for Mace4
10   assign(max_seconds, 60).
11 end_if.
12 
13 formulas(assumptions).
14 
15 sameBirthday(x,y) <-> (Twins(x,y) & -coincidence(x,y)) | (-Twins(x,y) & coincidence(x,y)).
16 all x(confident(x))->exists y(-coincidence(x,y)).
17 confident(John).
18 confident(x) -> exists y(sameBirthday(x,y)).
19 sameBirthday(John,Vasile).
20 
21 end_of_list.
22 
23 formulas(goals).
24 
25 end_of_list.
```

Explanation:

- solution of the problem: The professor had completely forgotten that two of the other students, who always sat next to each other, were identical twins.
- in the first line we defined a predicate sameBirthday(x,y), knowing that are 2 possibilities for the birthdays of 2 people in the class to be the same: even we speak about a pair of twins, and in the other case we speak about a coincidence. So we have to more predicates: Twins(x,y) and coincidence(x,y).
- in the next line it appears the predicate confident(x), meaning that x is confident and he didn't base on a coincidence. So, for every x that is confident, exists an y for which we do not have a coincidence(x,y).
- John is the student that was sure that are 2 people with the same birthday in the class, so we know that he is confident.
- in line 18, we expressed that if x is confident, for sure exists an y with the same birthday as x.

Results:

```
interpretation( 2, [number = 1,seconds = 0], [
    function(John, [0]),
    function(Vasile, [0]),
    function(f1(_), [0,0]),
    function(f2(_), [0,0]),
    function(f3(_), [0,0]),
    relation(confident(_), [1,0]),
    relation(Twins(_,_), [
        1,0,
        0,0]),
    relation(coincidence(_,_), [
        0,0,
        0,0]),
    relation(sameBirthday(_,_), [
        1,0,
        0,0])]).
```

1.5 References

-> https://www.ahapuzzles.com/logic/logic-puzzles/dinner-date/
-> https://www.brainzilla.com/logic/zebra/travel-agency/
-> https://www.ahapuzzles.com/logic/logic-puzzles/home-sick/
-> Smullyan, Raymond M book