Artificial intelligence - Project 2
- Propositional Logic and Predicates -

Baleanu Sorina-Diana

8/12/2020

1 PROBLEM 1: Black Friday

1.1 Clues of the problem

Five friends are side by side drinking juice and talking about the deals they got during the Black Friday sales. Follow the clues to find out who bought the laptop.
Clues:
1. The man drinking the Orange juice is exactly to the right of the man who got the 70 % discount.
2. Keith is 45 years old.
3. The man who bought the TV is exactly to the left of the man wearing the Red shirt.
4. At the third position is the man who got the 50 % discount.
5. Keith is next to the man wearing the White shirt.
6. The 25-year-old man is somewhere between the 35-year-old man and the 40-year-old man, in that order.
7. The man drinking Apple juice bought the Smartphone.
8. The 30-year-old man is exactly to the left of the man that bought the Beard trimmer.
9. Sean is the youngest.
10. The man that got the 40% discount is exactly to the right of the man who bought the Beard trimmer.
11. Keith is next to the 35-year-old man.
12. Eugene is 40 years old.
13. Sean is wearing the Black shirt.
14. At the fourth position is the man who got the biggest discount.
15. Dustin got 60% off.
16. The man drinking the Lemon juice is exactly to the right of the man drinking the Grape juice.
17. Keith bought a Game console.
18. The man who got the 80% discount is exactly to the left of the man who is wearing the Blue shirt.
19. The man drinking Grape juice bought the Beard trimmer.
20. The man wearing the Black shirt is somewhere to the right of Keith.
21. The man that bought the Smartphone is next to the man wearing the Black shirt.

1.2 Code implementation and solution

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).

%5 friends drinking juice
```

```
16
17  differentFrom(man1,man2).
18  differentFrom(man1,man3).
19  differentFrom(man1,man4).
20  differentFrom(man1,man5).
21
22  differentFrom(man2,man3).
23  differentFrom(man2,man4).
24  differentFrom(man2,man5).
25
26  differentFrom(man3,man4).
27  differentFrom(man3,man5).
28
29  differentFrom(man4,man5).
30
31  differentFrom(x,y) -> differentFrom(y,x).
32
33  %rightTo(x,y) - y is exactly to the right of x
34
35  rightTo(man1,man2).
36  rightTo(man2,man3).
37  rightTo(man3,man4).
38  rightTo(man4,man5).
39
40  -rightTo(man1,man1).
41  -rightTo(man1,man3).
42  -rightTo(man1,man4).
43  -rightTo(man1,man5).
44
45  -rightTo(man2,man1).
46  -rightTo(man2,man2).
47  -rightTo(man2,man4).
48  -rightTo(man2,man5).
49
50  -rightTo(man3,man1).
51  -rightTo(man3,man2).
52  -rightTo(man3,man3).
53  -rightTo(man3,man5).
54
55  -rightTo(man4,man1).
56  -rightTo(man4,man2).
57  -rightTo(man4,man3).
58  -rightTo(man4,man4).
59
60  -rightTo(man5,man1).
61  -rightTo(man5,man2).
62  -rightTo(man5,man3).
63  -rightTo(man5,man4).
64  -rightTo(man5,man5).
65
66  %leftTo(x,y) - y is exactly to the left of x
67  rightTo(y,x) <-> leftTo(x,y).
68
69  %nextTo(x,y) - y is to the left or to the right of x
```

```
rightTo(x,y) | leftTo(x,y) <-> nextTo(x,y).

%somewhereRight(x,y) - y is somewhere to the right of x

somewhereRight(man1,man2).
somewhereRight(man1,man3).
somewhereRight(man1,man4).
somewhereRight(man1,man5).

somewhereRight(man2,man3).
somewhereRight(man2,man4).
somewhereRight(man2,man5).

somewhereRight(man3,man4).
somewhereRight(man3,man5).

somewhereRight(man4,man5).

-somewhereRight(man1,man1).

-somewhereRight(man2,man1).
-somewhereRight(man2,man2).

-somewhereRight(man3,man1).
-somewhereRight(man3,man2).
-somewhereRight(man3,man3).

-somewhereRight(man4,man1).
-somewhereRight(man4,man2).
-somewhereRight(man4,man3).
-somewhereRight(man4,man4).

-somewhereRight(man5,man1).
-somewhereRight(man5,man2).
-somewhereRight(man5,man3).
-somewhereRight(man5,man4).
-somewhereRight(man5,man5).

%somewhereLeft(x,y) - y is somewhere to the left of x
somewhereRight(y,x) <-> somewhereLeft(x,y).

%somewhereInBetween (x,y,z) - y is somwhere in between x and z, x is to the left of y and z is to the ri
somewhereLeft(y,x) & somewhereRight(y,z) <-> somewhereInBetween (x,y,z).

%the colors of the men's t-shirts
black(x) | blue(x) | green(x) | red(x) | white(x).

%the name of the men
dustin(x) | eugene(x) | hank(x) | keith(x) | sean(x).

%the products purchased
beardtrimmer(x) | gameconsole(x) | laptop(x) | smartphone(x) | tv(x).

%the discounts
```

```
fortyp(x) | fiftyp(x) | sixtyp(x) | seventyp(x) | eightyp(x).

%the ages
twentyfivey(x) | thirtyy(x)| thirtyfivey(x) | fortyy(x) | fortyfivey(x).

%the juices men drink
apple(x) | cranberry(x)| grape(x) | lemon(x) | orange(x).

%each man has a different colored t-shirt
black(x) & black(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).

%each man has a different age
dustin(x) & dustin(y) -> -differentFrom(x,y).
eugene(x) & eugene(y) -> -differentFrom(x,y).
hank(x) & hank(y) -> -differentFrom(x,y).
keith(x) & keith(y) -> -differentFrom(x,y).
sean(x) & sean(y) -> -differentFrom(x,y).

%each man has purchased a different item
beardtrimmer(x) & beardtrimmer(y) -> -differentFrom(x,y).
gameconsole(x) & gameconsole(y) -> -differentFrom(x,y).
laptop(x) & laptop(y) -> -differentFrom(x,y).
smartphone(x) & smartphone(y) -> -differentFrom(x,y).
tv(x) & tv(y) -> -differentFrom(x,y).

%each man has a different discount
fortyp(x) & fortyp(y) -> -differentFrom(x,y).
fiftyp(x) & fiftyp(y) -> -differentFrom(x,y).
sixtyp(x) & sixtyp(y) -> -differentFrom(x,y).
seventyp(x) & seventyp(y) -> -differentFrom(x,y).
eightyp(x) & eightyp(y) -> -differentFrom(x,y).

%each man has a different age
twentyfivey(x) & twentyfivey(y) -> -differentFrom(x,y).
thirtyy(x) & thirtyy(y) -> -differentFrom(x,y).
thirtyfivey(x) & thirtyfivey(y) -> -differentFrom(x,y).
fortyy(x) &  fortyy(y) -> -differentFrom(x,y).
fortyfivey(x) & fortyfivey(y) -> -differentFrom(x,y).

%each man is drinking a different juice
apple(x) & apple(y) -> -differentFrom(x,y).
cranberry(x) & cranberry(y) -> -differentFrom(x,y).
grape(x) & grape(y) -> -differentFrom(x,y).
lemon(x) &  lemon(y) -> -differentFrom(x,y).
orange(x) & orange(y) -> -differentFrom(x,y).

%Clues

%1.The man drinking the Orange juice is exactly to the right of the man who got the 70 discount
rightTo(x,y) <-  orange(y) & seventyp(x) .
```

```
178
179 %2. Keith is 45 years old
180 keith(x) <-> fortyfivey(x).
181
182 %3.The man who bought the TV is exactly to the left of the man wearing the Red shirt.
183 leftTo(x,y) <-  tv(y) & red(x) .
184
185 %4.At the third position is the man who got the 50% discount
186 fiftyp(man3).
187
188 %5. Keith is next to the man wearing the White shirt.
189 nextTo(x,y) <- keith(x) & white(y).
190
191 %6. The 25-year-old man is somewhere between the 35-year-old man and the 40-year-old man, in that order.
192 somewhereInBetween(x,y,z) <-   thirtyfivey(x)& twentyfivey(y) & fortyy(z) .
193
194 %7. The man drinking Apple juice bought the Smartphone.
195 apple(x) <-> smartphone(x).
196
197 %8. The 30-year-old man is exactly to the left of the man that bought the Beard trimmer.
198 leftTo(x,y) <- beardtrimmer(x) & thirtyy(y) .
199
200 %9. Sean is the youngest.
201 sean(x) <-> twentyfivey(x).
202
203 %10. The man that got the 40% discount is exactly to the right of the man who bought the Beard trimmer.
204 rightTo(x,y) <- beardtrimmer(x) & fortyp(y) .
205
206 %11. Keith is next to the 35-year-old man.
207 nextTo(x,y) <- keith(x) & thirtyfivey(y).
208
209 %12. Eugene is 40 years old.
210 eugene(x) <-> fortyy(x).
211
212 %13. Sean is wearing the Black shirt.
213 sean(x) <-> black(x).
214
215 %14. At the fourth position is the man who got the biggest discount.
216 eightyp(man4).
217
218 %15. Dustin got 60% off.
219 dustin(x) <-> sixtyp(x).
220
221 %16. The man drinking the Lemon juice is exactly to the right of the man drinking the Grape juice
222 rightTo(x,y) <-  lemon(y) & grape(x) .
223
224 %17. Keith bought a Game console
225 keith(x) <-> gameconsole(x).
226
227 %18. The man who got the 80% discount is exactly to the left of the man who is wearing the Blue shirt.
228 leftTo(x,y) <- blue(x) & eightyp(y).
229
230 %19. The man drinking Grape juice bought the Beard trimmer.
231 grape(x) <-> beardtrimmer(x).
```

```
%20.The man wearing the Black shirt is somewhere to the right of Keith.
somewhereRight(x,y) <-  black(y) & keith(x) .

%21.The man that bought the Smartphone is next to the man wearing the Black shirt.
nextTo(x,y) <- smartphone(x) & black(y).

end_of_list.

formulas(goals).

end_of_list.
```

Explanation:

- For solving this logic puzzle I created the following predicates:
  The predicate differentFrom(x,y), expresses the idea that the men are distinct from one another. The "differentFrom" relation is symmetrical.
  The predicate rightTo(x,y), expresses the fact that the y is exactly to the right of x for every pair of men.
  The predicate leftTo(x,y), expresses the fact that the y is exactly to the left of x for every pair of men.
  The predicate nextTo(x,y), expresses the fact that the x is either to the left of y or to the right, this predicate was created using the rightTo and leftTo predicates.
  The predicate somewhereRight(x,y), expresses the fact that the y is somewhere to the right of x for every pair of men.
  The predicate somewhereLeft(x,y), expresses the fact that the y is somewhere to the right of x for every pair of men.
  The predicate somewhereInBetween(x,y,z), expresses the fact that the y is somewhere in between x and z, x being somewhere to the left of y and z somewhere to the right of y. The predicates black(x), blue(x), green(x), red(x), white(x) are used to define the colors of the men's t-shirts.
  The predicates dustin(x), eugene(x), hank(x), keith(x), sean(x) are used to define the names of the men.
  The predicates beardtrimmer(x), gameconsole(x), laptop(x), smartphone(x), tv(x)are used to define the product each man purchased.
  The predicates fortyp(x), fiftyp(x), sixtyp(x), seventyp(x), eightyp(x) are used to define the discounts.
  The predicates twentyfivey(x), thirtyy(x), thirtyfivey(x), fortyy(x) ,fortyfivey(x) are used to define the age of each man.
  The predicates apple(x) ,cranberry(x), grape(x), lemon(x), orange(x) are used to define the juice each man is drinking.
  We know that each man has a different colored t-shirt,a different name,has purchased a different product with a different discount, has a different age and is drinking a different juice , so I stated that using the predicate differentFrom and the predicates created for each attribute of the men, with statements like black(x) & black(y) -> -differentFrom(x,y).

Mace4 solution:

```
interpretation( 5, [number = 1,seconds = 0], [
    function(man1, [0]),
    function(man2, [1]),
    function(man3, [2]),
    function(man4, [3]),
    function(man5, [4]),
    relation(apple(_), [0,0,1,0,0]),
    relation(beardtrimmer(_), [0,0,0,1,0]),
    relation(black(_), [0,0,0,1,0]),
```

```
    relation(blue(_), [0,0,0,0,1]),
    relation(cranberry(_), [1,0,0,0,0]),
    relation(dustin(_), [0,1,0,0,0]),
    relation(eightyp(_), [0,0,0,1,0]),
    relation(eugene(_), [0,0,0,0,1]),
    relation(fiftyp(_), [0,0,1,0,0]),
    relation(fortyfivey(_), [1,0,0,0,0]),
    relation(fortyp(_), [0,0,0,0,1]),
    relation(fortyy(_), [0,0,0,0,1]),
    relation(gameconsole(_), [1,0,0,0,0]),
    relation(grape(_), [0,0,0,1,0]),
    relation(green(_), [1,0,0,0,0]),
    relation(hank(_), [0,0,1,0,0]),
    relation(keith(_), [1,0,0,0,0]),
    relation(laptop(_), [0,0,0,0,1]),
    relation(lemon(_), [0,0,0,0,1]),
    relation(orange(_), [0,1,0,0,0]),
    relation(red(_), [0,0,1,0,0]),
    relation(sean(_), [0,0,0,1,0]),
    relation(seventyp(_), [1,0,0,0,0]),
    relation(sixtyp(_), [0,1,0,0,0]),
    relation(smartphone(_), [0,0,1,0,0]),
    relation(thirtyfivey(_), [0,1,0,0,0]),
    relation(thirtyy(_), [0,0,1,0,0]),
    relation(tv(_), [0,1,0,0,0]),
    relation(twentyfivey(_), [0,0,0,1,0]),
    relation(white(_), [0,1,0,0,0]),
    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(leftTo(_,_), [
        0,0,0,0,0,
        1,0,0,0,0,
        0,1,0,0,0,
        0,0,1,0,0,
        0,0,0,1,0]),
    relation(nextTo(_,_), [
        0,1,0,0,0,
        1,0,1,0,0,
        0,1,0,1,0,
        0,0,1,0,1,
        0,0,0,1,0]),
    relation(rightTo(_,_), [
        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(somewhereLeft(_,_), [
        0,0,0,0,0,
        1,0,0,0,0,
```

```
        1,1,0,0,0,
        1,1,1,0,0,
        1,1,1,1,0]),
    relation(somewhereRight(_,_), [
        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(somewhereInBetween(_,_,_), [0,0,0,0,0,0,0,1,1,1,0,0,0,1,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0
```

Solution:

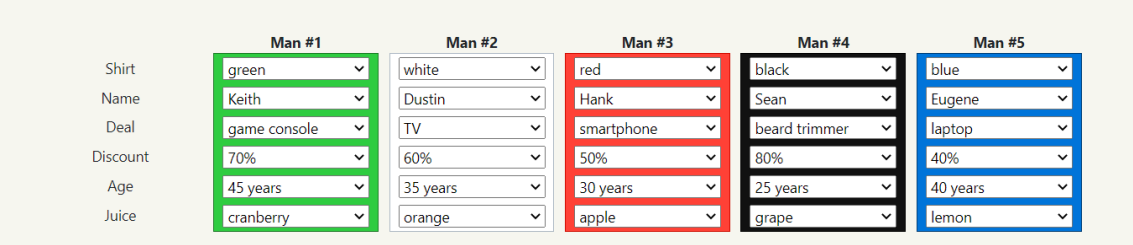

Figure 1: The solution of the logic puzzle

1.3 References

https://www.brainzilla.com/logic/zebra/black-friday/

2 PROBLEM 2: New York Beauty Salon

2.1 Clues of the problem

The New You Beauty Salon has a number of different hair colorings scheduled today. Using only the clues below, match each customer to her stylist, and determine the time of her appointment and the hair color dye she has chosen.
Clues:
1.The costumer that chose midnight blue is sometime before Georgia.
2. Susie's costumer has her appointment 4 hours after Tammie's client.
3.Whitney's costumer has her appointment 4 hours before Blanche.
4.Yvonne's costumer has her appointment sometime after the client that chose warm Mocha.
5. Yvonne's costumer, the client that chose honey blonde and Georgia are three different costumers.
6.Of Blanche and Rhonda's costumer one has the 4 pm appointment and the other one chose jet black.
7. The client tat chose jet black is either Opal's or Susie's client.
8.Georgia isn't helped by Rhonda.
9. Kelly doesn't have the 11 am appointment.
10. Of Opal's costumer and the client that chose vivacious red one has the 3pm appointment and the other one is Flora.
11. Alison , the costumer that chose brown sable, Susie's costumer, the costumer that chose honey blonde, the client that chose the 2pm appointment and Victoria 's customer are all different.
12. The costumer that chose honey blonde has her appointment 1 hour after the client that chose vivacious red.
13.Kelly didn't choose honey blonde.
14. Alison didn't choose deep copper.
15. Cristina doesn't have the 3pm appointment.

2.2 Code implementation and solution

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).

differentFrom(time11,time12).
differentFrom(time11,time13).
differentFrom(time11,time14).
```

```
differentFrom(time11,time15).
differentFrom(time11,time16).
differentFrom(time11,time17).

differentFrom(time12,time13).
differentFrom(time12,time14).
differentFrom(time12,time15).
differentFrom(time12,time16).
differentFrom(time12,time17).

differentFrom(time13,time14).
differentFrom(time13,time15).
differentFrom(time13,time16).
differentFrom(time13,time17).

differentFrom(time14,time15).
differentFrom(time14,time16).
differentFrom(time14,time17).

differentFrom(time15,time16).
differentFrom(time15,time16).

differentFrom(time16,time17).

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

%sometimeBefore(x,y) - y is sometime before x

sometimeBefore(time12,time11).

sometimeBefore(time13,time12).
sometimeBefore(time13,time11).

sometimeBefore(time14,time12).
sometimeBefore(time14,time11).
sometimeBefore(time14,time13).

sometimeBefore(time15,time14).
sometimeBefore(time15,time13).
sometimeBefore(time15,time12).
sometimeBefore(time15,time11).

sometimeBefore(time16,time15).
sometimeBefore(time16,time14).
sometimeBefore(time16,time13).
sometimeBefore(time16,time12).
sometimeBefore(time16,time11).

sometimeBefore(time17,time16).
sometimeBefore(time17,time15).
sometimeBefore(time17,time14).
sometimeBefore(time17,time13).
sometimeBefore(time17,time12).
sometimeBefore(time17,time11).
```

```
72
73  -sometimeBefore(time17,time17).
74
75  -sometimeBefore(time16,time17).
76  -sometimeBefore(time16,time16).
77
78  -sometimeBefore(time15,time17).
79  -sometimeBefore(time15,time16).
80  -sometimeBefore(time15,time15).
81
82  -sometimeBefore(time14,time17).
83  -sometimeBefore(time14,time16).
84  -sometimeBefore(time14,time15).
85  -sometimeBefore(time14,time14).
86
87  -sometimeBefore(time13,time13).
88  -sometimeBefore(time13,time14).
89  -sometimeBefore(time13,time15).
90  -sometimeBefore(time13,time16).
91  -sometimeBefore(time13,time17).
92
93  -sometimeBefore(time12,time12).
94  -sometimeBefore(time12,time13).
95  -sometimeBefore(time12,time14).
96  -sometimeBefore(time12,time15).
97  -sometimeBefore(time12,time16).
98  -sometimeBefore(time12,time17).
99
100 -sometimeBefore(time11,time11).
101 -sometimeBefore(time11,time12).
102 -sometimeBefore(time11,time13).
103 -sometimeBefore(time11,time14).
104 -sometimeBefore(time11,time15).
105 -sometimeBefore(time11,time16).
106 -sometimeBefore(time11,time17).
107
108 %sometimeAfter(x,y) - y is sometime after x
109 sometimeBefore(y,x) <-> sometimeAfter(x,y).
110
111 %fourAfter(x,y) - y is exactly four hours after x
112
113 fourAfter(time11,time15).
114 fourAfter(time12,time16).
115 fourAfter(time13,time17).
116
117 -fourAfter(time11,time11).
118 -fourAfter(time11,time12).
119 -fourAfter(time11,time13).
120 -fourAfter(time11,time14).
121 -fourAfter(time11,time16).
122 -fourAfter(time11,time17).
123
124 -fourAfter(time12,time11).
125 -fourAfter(time12,time12).
```

```
-fourAfter(time12,time13).
-fourAfter(time12,time14).
-fourAfter(time12,time15).
-fourAfter(time12,time17).

-fourAfter(time13,time11).
-fourAfter(time13,time12).
-fourAfter(time13,time13).
-fourAfter(time13,time14).
-fourAfter(time13,time15).
-fourAfter(time13,time16).

-fourAfter(time14,time11).
-fourAfter(time14,time12).
-fourAfter(time14,time13).
-fourAfter(time14,time14).
-fourAfter(time14,time15).
-fourAfter(time14,time16).
-fourAfter(time14,time17).

-fourAfter(time15,time11).
-fourAfter(time15,time12).
-fourAfter(time15,time13).
-fourAfter(time15,time14).
-fourAfter(time15,time15).
-fourAfter(time15,time16).
-fourAfter(time15,time17).

-fourAfter(time16,time11).
-fourAfter(time16,time12).
-fourAfter(time16,time13).
-fourAfter(time16,time14).
-fourAfter(time16,time15).
-fourAfter(time16,time16).
-fourAfter(time16,time17).

-fourAfter(time17,time11).
-fourAfter(time17,time12).
-fourAfter(time17,time13).
-fourAfter(time17,time14).
-fourAfter(time17,time15).
-fourAfter(time17,time16).
-fourAfter(time17,time17).

%fourBefore(x,y)(x,y) - y  is exactly four hours before x
fourAfter(y,x) <-> fourBefore(x,y).

%oneAfter(x,y) - y is exactly one hour after x

oneAfter(time11,time12).
oneAfter(time12,time13).
oneAfter(time13,time14).
oneAfter(time14,time15).
oneAfter(time15,time16).
```

```
oneAfter(time16,time17).

-oneAfter(time11,time11).
-oneAfter(time11,time14).
-oneAfter(time11,time13).
-oneAfter(time11,time15).
-oneAfter(time11,time16).
-oneAfter(time11,time17).

-oneAfter(time12,time11).
-oneAfter(time12,time12).
-oneAfter(time12,time14).
-oneAfter(time12,time15).
-oneAfter(time12,time16).
-oneAfter(time12,time17).

-oneAfter(time13,time11).
-oneAfter(time13,time12).
-oneAfter(time13,time13).
-oneAfter(time13,time15).
-oneAfter(time13,time16).
-oneAfter(time13,time17).

-oneAfter(time14,time11).
-oneAfter(time14,time12).
-oneAfter(time14,time13).
-oneAfter(time14,time14).
-oneAfter(time14,time16).
-oneAfter(time14,time17).

-oneAfter(time15,time11).
-oneAfter(time15,time12).
-oneAfter(time15,time13).
-oneAfter(time15,time14).
-oneAfter(time15,time15).
-oneAfter(time15,time17).

-oneAfter(time16,time11).
-oneAfter(time16,time12).
-oneAfter(time16,time13).
-oneAfter(time16,time14).
-oneAfter(time16,time15).
-oneAfter(time16,time16).

-oneAfter(time17,time11).
-oneAfter(time17,time12).
-oneAfter(time17,time13).
-oneAfter(time17,time14).
-oneAfter(time17,time15).
-oneAfter(time17,time16).
-oneAfter(time17,time17).

%the names of the costumers
alison(x) | blanche(x) | cristina(x) | flora(x) | georgia(x)| hope(x) | kelly(x).
```

```
234
235 %the names of the stylists
236 opal(x) | rohnda(x) | susie(x) | tammie(x) | victoria(x)| whitney(x) | yvonne(x).
237
238 %the colors
239 brownSable(x) | deepCopper(x) | honeyBlonde(x) | jetBlack(x) | midnightBlue(x)| vivaciousRed(x) | warmMo
240
241 %each costumer has an appointment at a different time
242 alison(x) & alison(y) -> -differentFrom(x,y).
243 blanche(x) & blanche(y) -> -differentFrom(x,y).
244 cristina(x) & cristina(y) -> -differentFrom(x,y).
245 flora(x) &  flora(y) -> -differentFrom(x,y).
246 georgia(x) & georgia(y) -> -differentFrom(x,y).
247 hope(x) &  hope(y) -> -differentFrom(x,y).
248 kelly(x) & kelly(y) -> -differentFrom(x,y).
249
250 %each stylist works at a different time
251 opal(x) & opal(y) -> -differentFrom(x,y).
252 rohnda(x) & rohnda(y) -> -differentFrom(x,y).
253 susie(x) & susie(y) -> -differentFrom(x,y).
254 tammie(x) &  tammie(y) -> -differentFrom(x,y).
255 victoria(x) & victoria(y) -> -differentFrom(x,y).
256 whitney(x) &  whitney(y) -> -differentFrom(x,y).
257 yvonne(x) & yvonne(y) -> -differentFrom(x,y).
258
259 %each stylist works at a different time
260 brownSable(x) & brownSable(y) -> -differentFrom(x,y).
261 deepCopper(x) & deepCopper(y) -> -differentFrom(x,y).
262 honeyBlonde(x) & honeyBlonde(y) -> -differentFrom(x,y).
263 jetBlack(x) &  jetBlack(y) -> -differentFrom(x,y).
264 midnightBlue(x) & midnightBlue(y) -> -differentFrom(x,y).
265 vivaciousRed(x) &  vivaciousRed(y) -> -differentFrom(x,y).
266 warmMocha(x) & warmMocha(y) -> -differentFrom(x,y).
267
268 %Clues
269
270 %1.The costumer that chose midnight blue is sometime before Georgia
271 sometimeBefore(x,y) <-  midnightBlue(y) & georgia(x) .
272
273 %2. Susie's costumer has her appointment 4 hours after Tammie's client
274 fourAfter(x,y) <- tammie(x) & susie(y).
275
276 %3.Whitney's costumer has her appointment 4 hours before Blanche.
277 fourBefore(x,y) <- whitney(y) & blanche(x).
278
279 %4.Yvonne's costumer has her appointment sometime after the client that chose warm Mocha.
280 sometimeAfter(x,y) <-  yvonne(y) & warmMocha(x) .
281
282 %5. Yvonne's costumer, the client that chose honey blonde and Georgia are three different costumers.
283
284 yvonne(x) & honeyBlonde(y) -> differentFrom(x,y).
285 yvonne(x) & georgia(y) -> differentFrom(x,y).
286 honeyBlonde(x) & georgia(y) -> differentFrom(x,y).
287
```

```
%6.Of Blanche and Rhonda's costumer one has the 4 pm appointment and the other one chose jet black.
(blanche(time16) &( rohnda(x)<-> jetBlack(x)))|(rohnda(time16) &( blanche(x)<-> jetBlack(x))).

%7. The client tat chose jet black is either Opal's or Susie's client.
(opal(x) <-> jetBlack(x))|(susie(x) <-> jetBlack(x)).

%8.Georgia isn't helped by Rhonda.
georgia(x) & rohnda(y) -> differentFrom(x,y).

%9. Kelly doesn't have the 11 am appointment
-kelly(time11).

%10. Of Opal's costumer and the client that chose vivacious red one has the 3pm appointment and the othe

(opal(time15)&(vivaciousRed(x)<->flora(x)))|(vivaciousRed(time15)&(opal(x)<->flora(x))).

%11. Alison , the costumer that chose brown sable, Susie's costumer, the costumer that chose honey blond
%the client that chose the 2pm appointment and victoria's customere are all different

-alison(time14).
-brownSable(time14).
-susie(time14).
-honeyBlonde(time14).
-victoria(time14).

alison(x) & brownSable(y) -> differentFrom(x,y).
alison(x) & susie(y) -> differentFrom(x,y).
alison(x) & honeyBlonde(y) -> differentFrom(x,y).
alison(x) & victoria(y) -> differentFrom(x,y).

%brownSable(x) & susie(y) -> differentFrom(x,y). - this staement is false
brownSable(x) & victoria(y) -> differentFrom(x,y).

susie(x) & honeyBlonde(y) -> differentFrom(x,y).

honeyBlonde(x) & victoria(y) -> differentFrom(x,y).

%12. The costumer that chose honey blonde has her appointment 1 hour after the client that chose vivacio
oneAfter(x,y) <- honeyBlonde(y) & vivaciousRed(x).

%13. Kelly didn't choose honey blonde.
kelly(x) & honeyBlonde(y) -> differentFrom(x,y).

%14. Alison didn't choose deep copper.
alison(x) & deepCopper(y) -> differentFrom(x,y).

%15. Cristina doesn't have the 3pm appointment
-cristina(time15).

end_of_list.

formulas(goals).

end_of_list.
```

Explanation:

- For solving this logic puzzle I created the following predicates:
  The predicate differentFrom(x,y), expresses the idea that the time of the appointments are distinct from one another. The "differentFrom" relation is symmetrical.
  The predicate sometimeBefore(x,y), express the fact that the appointment y is sometime before x,for each pair of appointments.
  The predicate sometimeAfter(x,y), expresses the fact that appointment y is sometime after x,for each pair of appointments.
  The predicate fourAfter(x,y), expresses the fact that the appointment y is exactly 4 hours after x,for each pair of appointments.
  The predicate fourBefore(x,y), express the fact that the appointment y is exactly 4 hours before x,for each pair of appointments.
  The predicate oneAfter(x,y), express the fact that the appointment y is exactly one hour after x,for each pair of appointments.

  The predicates alison(x), blanche(x), cristina(x), flora(x) ,georgia(x), hope(x), kelly(x) are used to define the names of the costumers.
  The predicates opal(x), rohnda(x), susie(x) ,tammie(x) ,victoria(x), whitney(x), yvonne(x) are used to define the names of the stylists.
  The predicates brownSable(x), deepCopper(x) ,honeyBlonde(x) ,jetBlack(x) ,midnightBlue(x), vivaciousRed(x) ,warmMocha(x) are used to define the color of hair dye each client chose.

  We know that every hour there is a different stylist working on a different client's appointment that has chosen a different hair dye , so I stated that using the predicate differentFrom and the predicates created for each atribute of the men, whit staements like alison(x) & alison(y) -> -differentFrom(x,y). The predicate differentFrom(x,y) was also used for clues 5,8,11,13 and 14.

  Clue 11 it's slightly mistaken. It states that Alison , the client that chose brown sable, Susie's costumer, the costumer that chose honey blonde,the client that has the 2pm appointment and Victoria 's costumer are all different, but the solution to the problem is that Susie dies her client's hair sable brown. If the statements of brownStable(x) are excluded the solution found by Mace4 isn't complete as it finds more possible options for the stylist Opal and none for Susie, same happens if the statements with susie(x) are excluded, so the only condition I excluded was that brownSable(x) & susie(x) -> -differentFrom(x,y) and with this condition excluded the program works properly.

Mace4 solution:

```
interpretation( 7, [number = 1,seconds = 0], [
    function(time11, [0]),
    function(time12, [1]),
    function(time13, [2]),
    function(time14, [3]),
    function(time15, [4]),
    function(time16, [5]),
    function(time17, [6]),
    relation(alison(_), [0,0,0,0,0,1,0]),
    relation(blanche(_), [0,0,0,0,1,0,0]),
    relation(brownSable(_), [0,0,0,0,0,0,1]),
    relation(cristina(_), [0,0,1,0,0,0,0]),
    relation(deepCopper(_), [0,0,0,1,0,0,0]),
    relation(flora(_), [0,1,0,0,0,0,0]),
    relation(georgia(_), [0,0,0,0,0,0,1]),
    relation(honeyBlonde(_), [0,0,1,0,0,0,0]),
```

```
    relation(hope(_), [1,0,0,0,0,0,0]),
    relation(jetBlack(_), [0,0,0,0,1,0,0]),
    relation(kelly(_), [0,0,0,1,0,0,0]),
    relation(midnightBlue(_), [0,0,0,0,0,1,0]),
    relation(opal(_), [0,0,0,0,1,0,0]),
    relation(rohnda(_), [0,0,0,0,0,1,0]),
    relation(susie(_), [0,0,0,0,0,0,1]),
    relation(tammie(_), [0,0,1,0,0,0,0]),
    relation(victoria(_), [0,1,0,0,0,0,0]),
    relation(vivaciousRed(_), [0,1,0,0,0,0,0]),
    relation(warmMocha(_), [1,0,0,0,0,0,0]),
    relation(whitney(_), [1,0,0,0,0,0,0]),
    relation(yvonne(_), [0,0,0,1,0,0,0]),
    relation(differentFrom(_,_), [
        0,1,1,1,1,1,1,
        1,0,1,1,1,1,1,
        1,1,0,1,1,1,1,
        1,1,1,0,1,1,1,
        1,1,1,1,0,1,0,
        1,1,1,1,1,0,1,
        1,1,1,1,0,1,0]),
    relation(fourAfter(_,_), [
        0,0,0,0,1,0,0,
        0,0,0,0,0,1,0,
        0,0,0,0,0,0,1,
        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]),
    relation(fourBefore(_,_), [
        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,
        1,0,0,0,0,0,0,
        0,1,0,0,0,0,0,
        0,0,1,0,0,0,0]),
    relation(oneAfter(_,_), [
        0,1,0,0,0,0,0,
        0,0,1,0,0,0,0,
        0,0,0,1,0,0,0,
        0,0,0,0,1,0,0,
        0,0,0,0,0,1,0,
        0,0,0,0,0,0,1,
        0,0,0,0,0,0,0]),
    relation(sometimeAfter(_,_), [
        0,1,1,1,1,1,1,
        0,0,1,1,1,1,1,
        0,0,0,1,1,1,1,
        0,0,0,0,1,1,1,
        0,0,0,0,0,1,1,
        0,0,0,0,0,0,1,
        0,0,0,0,0,0,0]),
    relation(sometimeBefore(_,_), [
```

```
71          0,0,0,0,0,0,0,
72          1,0,0,0,0,0,0,
73          1,1,0,0,0,0,0,
74          1,1,1,0,0,0,0,
75          1,1,1,1,0,0,0,
76          1,1,1,1,1,0,0,
77          1,1,1,1,1,1,0]]).
```

Solution:

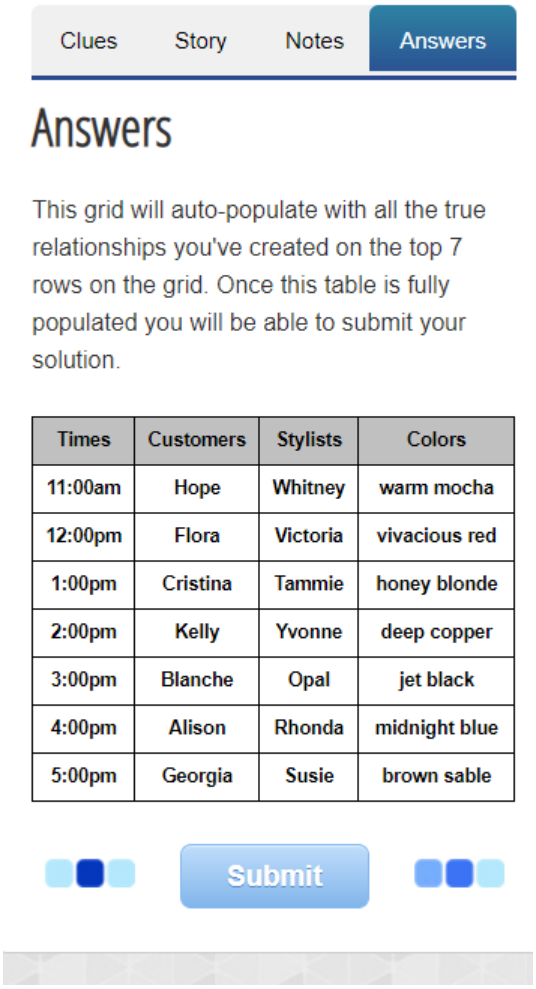

Figure 2: The solution of the logic puzzle

2.3 References

https://logic.puzzlebaron.com/play.php

3 PROBLEM 3: Blood Donation

3.1 Clues of the problem

Five women are side by side donating blood. Use logic and the clues to figure out their blood types.
Clues:
1.The A+ donor is next to the B+ donor.
2.Brooke is at one of the ends.
3.The woman wearing the Black shirt is somewhere to the left of the 150 lb woman.
4.The Actress is next to the Chef.
5. Kathleen is 40 years old.
6.The Florist is somewhere to the right of the woman wearing the Purple shirt.
7. The oldest donor weighs 130 lb.
8. Brooke is next to Nichole.
9. The 35-year-old woman is exactly to the left of the 30-year-old woman.
10. The 120 lb donor is somewhere between the the O- donor and the 150 lb donor, in that order.
11. Kathleen is at one of the ends.
12. The woman wearing the Purple shirt is somewhere to the right of the woman wearing the Green shirt.
13. The B+ donor weighs 140 lb.
14.The youngest woman is next to the 30-year-old woman.
15. The woman considered universal recipient is exactly to the left of the A+ donor.
16. Meghan is somewhere to the right of the woman wearing the Purple shirt.
17. The woman wearing the Green shirt is somewhere between the Actress and the woman wearing the Red shirt, in that order.
18. At one of the ends is the 130 lb woman.
19. The universal donor is 35 years old.
20.The Florist is somewhere between the Actress and the Engineer, in that order.
21.The woman wearing the Blue shirt is somewhere to the left of the woman wearing the Red shirt.
22.The AB+ donor is next to the youngest woman.

3.2 Code implementation and solution

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).

```

```
15 differentFrom(donnor1,donnor2).
16 differentFrom(donnor1,donnor3).
17 differentFrom(donnor1,donnor4).
18 differentFrom(donnor1,donnor5).
19
20 differentFrom(donnor2,donnor3).
21 differentFrom(donnor2,donnor4).
22 differentFrom(donnor2,donnor5).
23
24 differentFrom(donnor3,donnor4).
25 differentFrom(donnor3,donnor5).
26
27 differentFrom(donnor4,donnor5).
28
29 differentFrom(x,y) -> differentFrom(y,x).
30
31 %rightTo(x,y) - y is exactly to the right of x
32
33 rightTo(donnor1,donnor2).
34 rightTo(donnor2,donnor3).
35 rightTo(donnor3,donnor4).
36 rightTo(donnor4,donnor5).
37
38 -rightTo(donnor1,donnor1).
39 -rightTo(donnor1,donnor3).
40 -rightTo(donnor1,donnor4).
41 -rightTo(donnor1,donnor5).
42
43 -rightTo(donnor2,donnor1).
44 -rightTo(donnor2,donnor2).
45 -rightTo(donnor2,donnor4).
46 -rightTo(donnor2,donnor5).
47
48 -rightTo(donnor3,donnor1).
49 -rightTo(donnor3,donnor2).
50 -rightTo(donnor3,donnor3).
51 -rightTo(donnor3,donnor5).
52
53 -rightTo(donnor4,donnor1).
54 -rightTo(donnor4,donnor2).
55 -rightTo(donnor4,donnor3).
56 -rightTo(donnor4,donnor4).
57
58 -rightTo(donnor5,donnor1).
59 -rightTo(donnor5,donnor2).
60 -rightTo(donnor5,donnor3).
61 -rightTo(donnor5,donnor4).
62 -rightTo(donnor5,donnor5).
63
64 %leftTo(x,y) - y is exactly to the left of x
65 rightTo(y,x) <-> leftTo(x,y).
66
67 %nextTo(x,y) - y is to the left or to the right of x
68 rightTo(x,y) | leftTo(x,y) <-> nextTo(x,y).
```

```
69
70  %somewhereRight(x,y) - y is somewhere to the right of x
71
72  somewhereRight(donnor1,donnor2).
73  somewhereRight(donnor1,donnor3).
74  somewhereRight(donnor1,donnor4).
75  somewhereRight(donnor1,donnor5).
76
77  somewhereRight(donnor2,donnor3).
78  somewhereRight(donnor2,donnor4).
79  somewhereRight(donnor2,donnor5).
80
81  somewhereRight(donnor3,donnor4).
82  somewhereRight(donnor3,donnor5).
83
84  somewhereRight(donnor4,donnor5).
85
86  -somewhereRight(donnor1,donnor1).
87
88  -somewhereRight(donnor2,donnor1).
89  -somewhereRight(donnor2,donnor2).
90
91  -somewhereRight(donnor3,donnor1).
92  -somewhereRight(donnor3,donnor2).
93  -somewhereRight(donnor3,donnor3).
94
95  -somewhereRight(donnor4,donnor1).
96  -somewhereRight(donnor4,donnor2).
97  -somewhereRight(donnor4,donnor3).
98  -somewhereRight(donnor4,donnor4).
99
100 -somewhereRight(donnor5,donnor1).
101 -somewhereRight(donnor5,donnor2).
102 -somewhereRight(donnor5,donnor3).
103 -somewhereRight(donnor5,donnor4).
104 -somewhereRight(donnor5,donnor5).
105
106 %somewhereLeft(x,y) - y is somewhere to the left of x
107 somewhereRight(y,x) <-> somewhereLeft(x,y).
108
109 %somewhereInBetween (x,y,z) - y is somwhere in between x and z, x is to the left of y and z is to the ri
110 somewhereLeft(y,x) & somewhereRight(y,z) <-> somewhereInBetween (x,y,z).
111
112 %the colors of the donnors's t-shirts
113 black(x) | blue(x) | green(x) | purple(x) | red(x).
114
115 %the name of the donnors
116 andrea(x) | brooke(x) | kathleen(x) | meghan(x) | nichole(x).
117
118 %the blood type
119 a(x) | ab(x) | bp(x) | bn(x) | o(x).
120
121 %the weights
122 w120(x) | w130(x) | w140(x) | w150(x) | w160(x).
```

```
123
124 %the ages
125 twentyfivey(x) | thirtyy(x)| thirtyfivey(x) | fortyy(x) | fortyfivey(x).
126
127 %the jobs
128 actress(x) | chef(x)| engineer(x) | florist(x) | policewoman(x).
129
130 %each donnor has a different colored t-shirt
131 black(x) & black(y) -> -differentFrom(x,y).
132 blue(x) & blue(y) -> -differentFrom(x,y).
133 green(x) & green(y) -> -differentFrom(x,y).
134 red(x) & red(y) -> -differentFrom(x,y).
135 purple(x) & purple(y) -> -differentFrom(x,y).
136
137 %each donnor has a different name
138 andrea(x) & andrea(y) -> -differentFrom(x,y).
139 brooke(x) & brooke(y) -> -differentFrom(x,y).
140 kathleen(x) & kathleen(y) -> -differentFrom(x,y).
141 meghan(x) & meghan(y) -> -differentFrom(x,y).
142 nichole(x) & nichole(y) -> -differentFrom(x,y).
143
144 %each donnor has a different blood type
145 a(x) & a(y) -> -differentFrom(x,y).
146 ab(x) & ab(y) -> -differentFrom(x,y).
147 bp(x) & bp(y) -> -differentFrom(x,y).
148 bn(x) & bn(y) -> -differentFrom(x,y).
149 o(x) & o(y) -> -differentFrom(x,y).
150
151 %each donnoor has a different weight
152 w120(x) & w120(y) -> -differentFrom(x,y).
153 w130(x) & w130(y) -> -differentFrom(x,y).
154 w140(x) & w140(y) -> -differentFrom(x,y).
155 w150(x) & w150(y) -> -differentFrom(x,y).
156 w160(x) & w160(y) -> -differentFrom(x,y).
157
158 %each donnor has a different age
159 twentyfivey(x) & twentyfivey(y) -> -differentFrom(x,y).
160 thirtyy(x) & thirtyy(y) -> -differentFrom(x,y).
161 thirtyfivey(x) & thirtyfivey(y) -> -differentFrom(x,y).
162 fortyy(x) &  fortyy(y) -> -differentFrom(x,y).
163 fortyfivey(x) & fortyfivey(y) -> -differentFrom(x,y).
164
165 %each donnor has a different job
166 actress(x) & actress(y) -> -differentFrom(x,y).
167 chef(x) & chef(y) -> -differentFrom(x,y).
168 engineer(x) & engineer(y) -> -differentFrom(x,y).
169 florist(x) & florist(y) -> -differentFrom(x,y).
170 policewoman(x) & policewoman(y) -> -differentFrom(x,y).
171
172 %Clues
173
174 %1.The A+ donor is next to the B+ donor.
175 nextTo(x,y) <-  a(x) & bp(y) .
176
```

```
%2.Brooke is at one of the ends.
brooke(donnor1) | brooke(donnor5).

%3.The woman wearing the Black shirt is somewhere to the left of the 150 lb woman.
somewhereLeft(x,y) <-  black(y) & w150(x) .

%4.The Actress is next to the Chef.
nextTo(x,y) <-  actress(y) & chef(x) .

%5. Kathleen is 40 years old.
kathleen(x) <-> fortyy(x).

%6.The Florist is somewhere to the right of the woman wearing the Purple shirt.
somewhereRight(x,y) <-  florist(y) & purple(x) .

%7. The oldest donor weighs 130 lb.
fortyfivey(x) <-> w130(x).

%8. Brooke is next to Nichole.
nextTo(x,y) <- brooke(x) &  nichole(y) .

%9. The 35-year-old woman is exactly to the left of the 30-year-old woman.
leftTo(x,y) <- thirtyfivey(y) & thirtyy(x) .

%10. The 120 lb donor is somewhere between the the O- donor and the 150 lb donor, in that order.
somewhereInBetween(x,y,z) <-   o(x)& w120(y) & w150(z) .

%11. Kathleen is at one of the ends.
kathleen(donnor1)|kathleen(donnor5).

%12. The woman wearing the Purple shirt is somewhere to the right of the woman wearing the Green shirt.
somewhereRight(x,y) <-  purple(y) & green(x) .

%13. The B+ donor weighs 140 lb.
bp(x) <-> w140(x).

%14.The youngest woman is next to the 30-year-old woman.
nextTo(x,y) <- twentyfivey(x) &  thirtyy(y) .

%15. The woman considered universal recipient is exactly to the left of the A+ donor.
leftTo(x,y) <- a(x) & ab(y).

%16. Meghan is somewhere to the right of the woman wearing the Purple shirt.
somewhereRight(x,y) <-  meghan(y) & purple(x) .

%17. The woman wearing the Green shirt is somewhere between the Actress and the woman wearing the Red sh
somewhereInBetween(x,y,z) <-   actress(x)& green(y) & red(z) .

%18. At one of the ends is the 130 lb woman.
w130(donnor1) | w130(donnor5).

%19. The universal donor is 35 years old.
o(x) <-> thirtyfivey(x).
```

```
%20.The Florist is somewhere between the Actress and the Engineer, in that order.
somewhereInBetween(x,y,z) <-   actress(x)& florist(y) & engineer(z) .

%21.The woman wearing the Blue shirt is somewhere to the left of the woman wearing the Red shirt.
somewhereLeft(x,y) <-  blue(y) & red(x) .

%22.The AB+ donor is next to the youngest woman.
nextTo(x,y) <- ab(x) &  twentyfivey(y) .

end_of_list.

formulas(goals).

end_of_list.
```

Explanation:

- For solving this logic puzzle I created the following predicates:
  The predicate differentFrom(x,y), express the idea that the donors are distinct from one another. The "differentFrom" relation is symmetrical.
  The predicate rightTo(x,y), express the fact that the y is exactly to the right of x, for every pair of donors.
  The predicate leftTo(x,y), express the fact that the y is exactly to the left of x, for every pair of donors.
  The predicate nextTo(x,y), express the fact that the x is eighter to the left of y or to the right, this predicate was created using the rightTo and leftTo predicates.
  The predicate somewhereRight(x,y), express the fact that the y is somewhere to the right of x for every pair of donors.
  The predicate somewhereLeft(x,y), express the fact that the y is somewhere to the right of x for every pair of donors.
  The predicate somewhereInBetween(x,y,z), express the fact that the y is somewhere in between x and z, x being somewhere to the left of y and z somewhere to the right of y. The predicates black(x) ,blue(x) , green(x) ,purple(x) , red(x) are used to define the colors of the donors 's t-shirts.
  The predicates andrea(x) ,brooke(x) ,kathleen(x) , meghan(x) , nichole(x) used to define the names of the donors.
  The predicates a(x) , ab(x) , bp(x) , bn(x) , o(x) are used to define the blood types of the donors.
  The predicates w120(x) , w130(x) , w140(x) , w150(x) , w160(x) are used to define the weight of each donor.
  The predicates twentyfivey(x), thirtyy(x), thirtyfivey(x), fortyy(x) ,fortyfivey(x) are used to define the age of each donor.
  The predicates actress(x) ,chef(x), engineer(x), florist(x), policewoman(x) are used to define the job of each donor.
  We know that each donor has a different colored t-shirt,a different name,has a different blood type, a different weight, has a different age and has a different job , so I stated that using the predicate differentFrom and the predicates created for each atribute of the men, with staements like black(x) & black(y) -> -differentFrom(x,y).

Mace4 solution:

```
interpretation( 5, [number = 1,seconds = 1], [
    function(donnor1, [0]),
    function(donnor2, [1]),
    function(donnor3, [2]),
    function(donnor4, [3]),
    function(donnor5, [4]),
    relation(a(_), [0,0,0,1,0]),
```

```
    relation(ab(_), [0,0,1,0,0]),
    relation(actress(_), [1,0,0,0,0]),
    relation(andrea(_), [0,0,1,0,0]),
    relation(black(_), [1,0,0,0,0]),
    relation(blue(_), [0,0,0,1,0]),
    relation(bn(_), [1,0,0,0,0]),
    relation(bp(_), [0,0,0,0,1]),
    relation(brooke(_), [1,0,0,0,0]),
    relation(chef(_), [0,1,0,0,0]),
    relation(engineer(_), [0,0,0,0,1]),
    relation(florist(_), [0,0,0,1,0]),
    relation(fortyfivey(_), [1,0,0,0,0]),
    relation(fortyy(_), [0,0,0,0,1]),
    relation(green(_), [0,1,0,0,0]),
    relation(kathleen(_), [0,0,0,0,1]),
    relation(meghan(_), [0,0,0,1,0]),
    relation(nichole(_), [0,1,0,0,0]),
    relation(o(_), [0,1,0,0,0]),
    relation(policewoman(_), [0,0,1,0,0]),
    relation(purple(_), [0,0,1,0,0]),
    relation(red(_), [0,0,0,0,1]),
    relation(thirtyfivey(_), [0,1,0,0,0]),
    relation(thirtyy(_), [0,0,1,0,0]),
    relation(twentyfivey(_), [0,0,0,1,0]),
    relation(w120(_), [0,0,1,0,0]),
    relation(w130(_), [1,0,0,0,0]),
    relation(w140(_), [0,0,0,0,1]),
    relation(w150(_), [0,0,0,1,0]),
    relation(w160(_), [0,1,0,0,0]),
    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(leftTo(_,_), [
        0,0,0,0,0,
        1,0,0,0,0,
        0,1,0,0,0,
        0,0,1,0,0,
        0,0,0,1,0]),
    relation(nextTo(_,_), [
        0,1,0,0,0,
        1,0,1,0,0,
        0,1,0,1,0,
        0,0,1,0,1,
        0,0,0,1,0]),
    relation(rightTo(_,_), [
        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(somewhereLeft(_,_), [
```

```
        0,0,0,0,0,
        1,0,0,0,0,
        1,1,0,0,0,
        1,1,1,0,0,
        1,1,1,1,0]),
    relation(somewhereRight(_,_), [
        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(somewhereInBetween(_,_,_), [0,0,0,0,0,0,0,1,1,1,0,0,0,1,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
```

Solution:

	Donor #1	Donor #2	Donor #3	Donor #4	Donor #5
Shirt	black	green	purple	blue	red
Name	Brooke	Nichole	Andrea	Meghan	Kathleen
Blood		O-	AB+	A+	B+
Age	45 years	35 years	30 years	25 years	40 years
Weight	130 lb	160 lb	120 lb	150 lb	140 lb
Job	actress	chef	policewoman	florist	engineer

Figure 3: The solution of the logic puzzle

3.3 References

https://www.brainzilla.com/logic/zebra/blood-donation/