Department of Computer Science
Technical University of Cluj-Napoca

Puzzle

Position in department store [1]

In a certain department store the position of buyer, cashier, clerk, floorwalker, and manager are held, though not necessarily respectively, by Miss Ames, Miss Brown, Mr. Conroy, Mr. Davis, and Mr. Evans.

The cashier and the manager were roommates in college.

The buyer is a bachelor.

Evans and Miss Ames have had only business contacts with each other.

Mrs. Conroy was greatly disappointed when her husband told her that the manager had refused to give him a raise.

Davis is going to be the best man when the clerk and the cashier are married.

What position does each person hold?

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

set(ignore_option_dependencies). % GUI handles dependencies

if(Prover9). % Options for Prover9
  assign(max_seconds, 120).
  assign(max_weight, 10000).
end_if.

if(Mace4).   % Options for Mace4
  assign(domain_size, 5).
  assign(start_size, 5).
  assign(end_size, 5).
  assign(max_models, 100000).
  assign(max_seconds, 60).
end_if.

formulas(assumptions).

hasJob(x, buyer) | hasJob(x, cashier) | hasJob(x,clerk) | hasJob(x,floorwalker) | hasJob(x
hasJob(Ames, x) | hasJob(Brown, x) | hasJob(Conroy, x) | hasJob(Davis, x) | hasJob(Evans,

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

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

hasJob(x,y) & hasJob(z,y) -> -diffPeople(x,z).
hasJob(x,y) & hasJob(x,z) -> -diffJobs(y,z).

(male(x) & -female(x)) | (female(x) & -male(x)).

couple(x,y)->(male(x) & female(y)) | (male(y) & female(x)).
couple(x,y)->couple(y,x).
couple(x,y)-> diffPeople(x,y).
```

```

bachelor(x) -> -couple(x,y) & male(x).
onlyBusinessContact(x,y)->onlyBusinessContact(y,x).
onlyBusinessContact(x,y) -> -couple(x,y).
couple(x,y) -> -onlyBusinessContact(x,y).

diffJobs(buyer,cashier).
diffJobs(buyer,clerk).
diffJobs(buyer,floorwalker).
diffJobs(buyer,manager).
diffJobs(cashier,clerk).
diffJobs(cashier,floorwalker).
diffJobs(cashier,manager).
diffJobs(clerk,floorwalker).
diffJobs(clerk,manager).
diffJobs(floorwalker,manager).

diffPeople(Ames,Brown).
diffPeople(Ames,Conroy).
diffPeople(Ames,Davis).
diffPeople(Ames,Evans).
diffPeople(Brown,Conroy).
diffPeople(Brown,Davis).
diffPeople(Brown,Evans).
diffPeople(Conroy,Davis).
diffPeople(Conroy,Evans).
diffPeople(Davis,Evans).

female(Ames) & female(Brown) & male(Conroy) & male(Davis) & male(Evans).
hasJob(x, cashier) & hasJob(y, manager) -> -onlyBusinessContact(x,y).
hasJob(x, buyer) -> bachelor(x).
onlyBusinessContact(Evans,Ames).
onlyBusinessContact(Ames,Evans).
-bachelor(Conroy).

-hasJob(Conroy, manager).
-hasJob(Conroy, buyer).
-hasJob(Conroy,clerk).
-hasJob(Conroy,cashier).

hasJob(x,clerk) & hasJob(y,cashier) -> couple(x,y).
-hasJob(Davis, clerk).
-hasJob(Davis, cashier).
hasJob(x,clerk) & hasJob(y,cashier)-> -onlyBusinessContact(x,Davis) | -onlyBusinessContact

end_of_list.

formulas(goals).

hasJob(Ames,manager) & hasJob(Brown,cashier) & hasJob(Conroy,floorwalker) & hasJob(Davis,b

end_of_list.
```

Solutions

First we start with Conroy -¿ cannot be manager, buyer (because is not a bachelor), clerk or cashier (because is already married) =¿ Conroy is the floorwalker.

Next, Davis is a bachelor because one man is going to marry and the other one is already married =¿ Davis is the buyer.

Evans will marry with Brown because Evans and Ames have had only business contacts =¿ Ames is the manager.

Ames roommate with cashier =¿ Brown is the cashier.

Evans is the clerk.

Knights and Knaves and Spies [1]

On the island of knights and knaves and spies, you are approached by three people wearing different colored clothes.

You know that one is a knight, one is a knave, and one is a spy. They speak in the following order:

The man wearing blue says, ”I am a knight.”
The man wearing red says, ”He speaks the truth.”
The man wearing green says, ”I am a spy.”

Who is the knight, who is the knave, and who is the spy?

```
% 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).

knight(x)-> -knave(x) & -spy(x).
knave(x) -> -knight(x) & -spy(x).
spy(x) -> -knight(x) & -knave(x).
-knight(x) & -knave(x) -> spy(x).

knight(A) | knave(A) | spy(A).
knight(B) | knave(B) | spy(B).
knight(C) | knave(C) | spy(C).

knight(A) -> -knight(B) & -knight(C).
knight(B) -> -knight(A) & -knight(C).
knight(C) -> -knight(B) & -knight(A).

knave(A) -> -knave(B) & -knave(C).
knave(B) -> -knave(A) & -knave(C).
knave(C) -> -knave(B) & -knave(A).

spy(A) -> -spy(B) & -spy(C).
spy(B) -> -spy(A) & -spy(C).
spy(C) -> -spy(B) & -spy(A).

%I am a knight.
knight(A) -> knight(A).
knave(A) -> -knight(A).
spy(A) -> knight(A) | -knight(A).

knight(B) -> knight(A).
knave(B) -> -knight(A).
```

```
spy(B) -> knight(A) | -knight(A).

knight(C) -> spy(C).
knave(C) -> -spy(C).
spy(C) -> spy(C) | -spy(C).

end_of_list.

formulas(goals).

knight(A) & spy(B) & knave(C).

end_of_list.
```

Solutions

Assume Blue is knave results Red is spy and Green is knight but the 3 proposition say is a spy.

Assume Blue is knight results Red is spy and Green is knave.

Hint: Do not assume somebody is a spy.

BEA O COLA [1]

In the figure below we see a multiplication. Letters represents digits and all letters are distinct digits. Find a solution.

$$\begin{array}{cccc} & B & E & A \\ \times & & & O \\ \hline C & O & L & A \end{array}$$

Figure 1: Multiplication.

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

set(ignore_option_dependencies). % GUI handles dependencies

if(Prover9). % Options for Prover9
  assign(max_seconds, 609).
  set(prolog_style_variables).
end_if.

if(Mace4).   % Options for Mace4
  assign(domain_size, 100).
  assign(start_size, 100).
  assign(end_size, 100).
  assign(max_models, 100).
  assign(max_seconds, 600).
  set(prolog_style_variables).
end_if.

if(Mace4).   % Additional input for Mace4
set(integer_ring).
end_if.

formulas(assumptions).

f(X) -> ((X=0) | (X=1) | (X=2) | (X=3) | (X=4) | (X=5) | (X=6) | (X=7) | (X=8) | (X=9)).
f(0).
f(1).
f(2).
f(3).
f(4).
f(5).
f(6).
f(7).
f(8).
f(9).

f(a).
f(b).
f(c).
f(e).
f(l).
f(o).

f(c1).
f(c2).
```

```
f(c3).
c2 != 10.
a * o = (c1 * 10) + a.
(e * o) + c1 = ((c2 * 10) + l).
(b * o) + c2 = ((c3 * 10) + o).
c = c3.

b != 0.
o != 0.
c != 0.

%diff(X,Y)-> X!=Y.
%diff(a,b).
%diff(a,c).
%diff(a,e).
%diff(a,o).
%diff(a,l).
%diff(b,c).
%diff(b,e).
%diff(b,o).
%diff(b,l).
%diff(c,e).
%diff(c,o).
%diff(c,l).
%diff(e,o).
%diff(e,l).
%diff(o,l).
a!=b.
a!=c.
a!=e.
a!=l.
a!=o.
b!=c.
b!=e.
b!=l.
b!=o.
c!=e.
c!=l.
c!=o.
e!=l.
e!=o.
l!=o.

end_of_list.

formulas(goals).

end_of_list.
```

Solutions

First we exclude the obvious. B!=0, O!=0 and 1, C!=0 and 9.
Assume O is 6 results A=2 reminder 1.
Assume E is 3 result L=9 reminder need to be even contradiction.
Assume E is 4 results L=5 reminder 2.
(B * 6 + 2) % 10 = 6 results B is 4 and C is 2 contradiction with A.
B can be 9 and C=5 contradiction with L.

Assume E is 7 results L=3 reminder 4
can not choose B.

Assume A=0.
Assume O is 6.
E can be 7 or 8. Assume E is 7 results L=2.
(B * 6 + 4) % 10 = 6 results B=2 and C=1.

270 * 6 = 1620.

Models	a	b	c	e	l	o
1	0	2	1	3	4	8
2	0	7	2	8	4	3
...	...	...	...	...	...	...
15	...	...	...	...	...	...

Six coin problem [1]

A well-known example has up to six items, say coins (or balls), that are identical in weight except one, which is lighter than the others—a counterfeit (an oddball). The difference is perceptible only by weighing them on scale—but only the coins themselves can be weighed.

How can one isolate the counterfeit coin with only two weightings?

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

set(ignore_option_dependencies). % GUI handles dependencies

if(Prover9). % Options for Prover9
  assign(max_seconds, 609).
  set(prolog_style_variables).
end_if.

if(Mace4).   % Options for Mace4
  assign(domain_size, 7).
  assign(start_size, 7).
  assign(end_size, 7).
  assign(max_models, 100).
  set(prolog_style_variables).
  set(integer_ring).
  assign(max_seconds, 60).
end_if.

if(Mace4).   % Additional input for Mace4
set(integer_ring).
end_if.

formulas(assumptions).

a|b|c|d|e|f.
a -> -b & -c & -d & -e & -f.
b -> -a & -c & -d & -e & -f.
c -> -b & -a & -d & -e & -f.
d -> -b & -c & -a & -e & -f.
e -> -b & -c & -d & -a & -f.
f -> -b & -c & -d & -e & -a.

a|b|c|d|e|f -> ((a|b|c -> (-d)&(-e)&(-f)) | (d|e|f -> (-a)&(-b)&(-c))).

a|b|c ->((a|b-> -c) | (-(a|b)->c & -a & -b)).
e|f|g ->((e|f-> -g) | (-(e|f)->g & -e & -f)).

a|b -> ((-a->b) | (-b->a)).
e|f -> ((-e->f) | (-f->e)).

end_of_list.

formulas(goals).

end_of_list.
```

Solutions

First, we scale 3 coins with 3.
We choose the lighter group.
Next, we will take 2 random coins from the lighter group and scale them.
If one is heavier the other is lighter. (Answer)
If are equals the third coin is the lightest.(Answer)
Problem solve in 2 weightings.

Bibliography

[1] Boris A Kordemsky. *The Moscow puzzles: 359 mathematical recreations.* Courier Corporation, 1992.