

Werewolves (**wolves**)

— [Run LPL Code](#) , [HTML Document](#) —

Problem: Suppose you are visiting a forest in which every inhabitant is either a knight or a knave. Knights always tell the truth and knaves always lie. In addition some of the inhabitants are werewolves and have the annoying habit of sometimes turning into wolves at night and devouring people. A werewolf can be either a knight or a knave.

You are interviewing three inhabitants, *A*, *B*, and *C*, and it is known that exactly one of them is a werewolf. They make the following statements:

```
A: I am a werewolf.  
B: I am a werewolf.  
C: At most one of us is a knight.
```

Give a complete classification of *A*, *B* and *C*. (see [3] and [1]).

Modeling Steps

We have three persons (a set p). Each is either a knight or a knave, and some of them can turn into werewolves.

1. We introduce a binary variable x_p for each person which is 1 if the person is a knight, otherwise it is 0 (and the person is then a knave).
2. For each person we introduce another variable y_p , which is 1 if the person is a werewolf otherwise it is zero.
3. There is exactly one werewolf, hence $\bigvee_p y_p$ (or $\sum_p y_p = 1$).
4. Person *A* says to be a werewolf. If he is a knight then he is a werewolf and if he is a knave then he is not a werewolf, that means: $x_1 \leftrightarrow y_1$ (or $x_1 = y_1$).
5. For the second person *B* the same condition holds: $x_2 \leftrightarrow y_2$ (or $x_2 = y_2$).
6. The third person *C* says, that there are at most one knight. If he is a knight then he tells the truth otherwise his statement is not true. Hence *C* being a knight is equivalent to $\text{atmost}(1)_p x_p$ ¹ In LPL syntax this can be written as `atmost(1) {p} x[p]`. Hence the constraint can be formulated as:

$$x_3 \leftrightarrow \text{atmost}(1)_p x_p$$

7. We minimize x_3 to check whether *C* is a knight.

Listing 1: The Complete Model implemented in LPL [4]

```
model wolves "Werewolves";  
set p := [A B C] "Persons";  
binary variable  
  x{p} "The person p is a knight";  
  y{p} "The person p is a werewolf";  
constraint
```

¹The operator $\text{atmost}(1)_p x_p$ means “at most 1 x_p is true, or: $\sum_p x_p \geq 1$ ”

```

A: xor{p} y "Exactly one is a werewolf";
Asays: x[1] <-> y[1];
Bsays: x[2] <-> y[2];
Csays: x[3] <-> atleast(1){p} x;
minimize Cknight: x[3];
Write{p}('%s_is_a_%5s\n', p, if (x, 'knight', 'knave'));
Write{p|y}('%s_is_a_werewolf\n', p);
end

```

Solution: C is a knight because $x_3 = 1$ in the solution, since x_3 was minimized C cannot be a knave. A and B are both knaves and not werewolves. Minimizing y_3 shows that C is also the only werewolf.

Further Comments: The constraint $x_1 \leftrightarrow y_1$ is translated by LPL into the two linear inequalities $x_1 \leq y_1$ and $x_1 \geq y_1$ which is the same as $x_1 = y_1$.

The constraint $x_3 \leftrightarrow \text{atmost}(1)_p x_p$ is slightly more complicated. It is translated into the linear inequalities $2 \leq x_A + x_B + 3x_C \leq 3$. To verify that this is a correct translation, one may construct the true table for the expressions (note: 0=false, 1=true). The table shows that the two expressions are equivalent:

x_C	x_A	x_B	$x_C \leftrightarrow \text{atmost}(1)_p x_p$	$2 \leq x_A + x_B + 3x_C \leq 3$
1	1	1	<i>false</i>	<i>false</i>
1	1	0	<i>false</i>	<i>false</i>
1	0	1	<i>false</i>	<i>false</i>
1	0	0	<i>true</i>	<i>true</i>
0	1	1	<i>true</i>	<i>true</i>
0	1	0	<i>false</i>	<i>false</i>
0	0	1	<i>false</i>	<i>false</i>
0	0	0	<i>false</i>	<i>false</i>

Questions

1. Is this the only solution?
2. Suppose, A says that he is not a werewolf. How does the solution change?
3. Suppose that the only knight is a werewolf. Furthermore, A and B make the following statement:

```

A: At least one of the three of us is a knave.
B: C is a knight.

```

Who is the werewolf now?

Answers

1. Yes! minimize x_3 shows that C must be a knight, maximizing x_2 shows that person B must be a knave, and maximizing x_1 shows that A must also be a knave. Finally, minimizing y_3 shows that C is the unique werewolf.
2. The only thing that changes is, that now A is the werewolf. The second constraint must be replaced by

```

constraint Asays: x[1] <-> ~y[1];

```

3. A is the werewolf. The complete model is as follows:

```
model wolves1 "Werewolves II";
set p := [A B C] "Persons";
binary variable x{p}; y{p};
constraint
  A: xor{p} y;
  B{p}: y -> x "The werewolf is a knight";
  Asays: x[1] <-> or{p} ~x "At least one is a knave, says A";
  Bsays: x[2] = x[3] "B says: C is a knight";
minimize any: y[1];
Write{p}('%s_is_a_%5s\n', p, if (x, 'knight', 'knave'));
Write{p|y}('%s_is_a_werewolf\n', p);
end
```

The model can be downloaded at [wolves1](#).

References

- [1] Nolan Clare. <http://www.chlond.demon.co.uk/academic/puzzles.html>.
- [2] MatMod. Homepage for Learning Mathematical Modeling : <https://matmod.ch>.
- [3] Smullyan R. *The Lady or The Tiger*. Oxford University Press, 1991.
- [4] Hürlimann T. Reference Manual for the LPL Modeling Language, most recent version. <https://matmod.ch/lpl/doc/manual.pdf>.