IP Modeling and the Logical Puzzles of Raymond Smullyan

Martin J. Chlond and Cath M.Toase
Lancashire Business School
University of Central Lancashire
Preston, PR1 2HE, UK

mchlond@uclan.ac.uk
cmtoase1@uclan.ac.uk

ABSTRACT

The ability to include logical conditions within Integer Programming (IP) models has many applications in OR/MS. Although the modeling of logical conditions in IP is simple in principle, in actual practice the exercise can be quite painstaking and prone to error. To become adept therefore it is necessary for practitioners to be well drilled. This paper presents the puzzles of Raymond Smullyan as a rich source of examples for the instructor that offer all the pedagogical features of more conventional text book examples but with added flavors of whimsy and caprice.

Editor's note: This is a pdf copy of an html document which resides at
http://ite.pubs.informs.org/Vol3No3/ChlondToase/

1 INTRODUCTION

From the late 1970s to the late 1990s, Raymond Smullyan, a professor of Mathematics and Philosophy from New York, published several books containing an eclectic mix of riddles, logic puzzles and brain teasers to appeal to adults and children alike (Smullyan, 1978, 1979, 1981, 1982, 1982, 1985, 1987, 1992, 1997). The puzzles are at once quaint and challenging and many are embedded in scenarios taken from popular literature and folklore such as Alice in Wonderland (Smullyan, 1982), The Arabian Knights (Smullyan, 1981) and Sherlock Holmes (Smullyan, 1979).

Aside from his respected academic work Smuyllans career spans that of musician, writer, humorist and even childrens magician. The charisma and charm of his writing has introduced many newcomers to the pleasure of mental puzzles.

In this paper we select a range of Smullyans logic puzzles and demonstrate how modeling logical conditions using IP can be applied to produce solutions. The puzzles we have chosen include some that are solvable with a moments reflection to one where it seems impossible to know where to start (Logical labyrinth Section 2.2).

The importance of making learning fun was emphasized in a previous paper where we demonstrated the applicability of IP as a means of solving chessboard placement puzzles (Chlond, and Toase, 2002). As we continue to look for ways to encourage those students with little confidence in concepts they perceive to be mathematical we found the discovery of the applicability of IP to a whole new problem type to be exciting. Certainly, in our teaching experience so far, student response has been very positive.

2 THE PUZZLES

2.1 Knights, knaves and werewolves

The first two puzzles are taken from What is the Name of this Book? (Smullyan, 1978). 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 knight and devouring people. A werewolf can be either a knight or a knave.

Werewolves II

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.

Werewolves IV

This time we get the following statements:

  ISSN: 1532-0545

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

Given that there is exactly one werewolf and that he is a knight, who is the werewolf?

2.2 Ladies or Tigers?

The next two puzzles are taken from The Lady or the Tiger (Smullyan, 1982). The relevant chapter, Ladies or Tigers, contains 12 puzzles of increasing difficulty. In each puzzle a prisoner is faced with a decision where he must open one of several gdoors. In the first few examples each room contains either a lady or a tiger and in the more difficult examples rooms may also be empty. We have chosen to include one of the simplest and the most difficult.

If the prisoner opens a door to find a lady he will marry her and if he opens a door to find a tiger he will be eaten alive. We assume that the prisoner would prefer to be married than eaten alive. It is also assumed that the lady is in some way special to the prisoner and he would prefer to find and marry her rather than an open a door into and empty room.

Each of the doors has a sign bearing a statement that may be either true or false.

The Second Trial

This puzzle involves two rooms.

The statement on door one says, "At least one of these rooms contains a lady."

The statement on door two says, "A tiger is in the other room."

The statements are either both true or both false.

A Logical Labyrinth

The final puzzle in this section of the book involves nine rooms. The statements on the nine doors are:

Door1 The lady is in an odd-numbered room
Door2 This room is empty
Door3 Either sign 5 is right or sign 7 is wrong
Door4 Sign 1 is wrong
Door5 Either sign 2 or sign 4 is right
Door6 Sign 3 is wrong
Door7 The lady is not in room 1
Door8 This room contains a tiger and room 9 is empty
Door9 This room contains a tiger and sign 6 is wrong

In addition, the prisoner is informed that only one room contains a lady; each of the others either contains a tiger or is empty. The sign on the door of the room containing the lady is true, the signs on all the doors containing tigers are false, and the signs on the doors of empty rooms can be either true or false.

The puzzle as stated does not have a unique solution until the prisoner is told whether or not room eight is empty and this knowledge enables him to find a unique solution.

3 MODELING TOOLS

3.1 Indicator variables

It is useful to develop linear constraints to force an indicator variable to 1 if and only if a particular proposition is true. Four examples are presented as follows.

In each case $x = [x_1, x_2, .., x_c]$, F_x is a linear function of x and U and L are upper and lower bounds respectively on F_x.

$\delta = 1$ if $F_x \geq n$, 0 otherwise

$$F_x - (U - n + 1)\delta \leq n - 1 \quad (1)$$

$$F_x - (n - L)\delta \geq L \quad (2)$$

$\delta = 1$ if $F_x \leq n$, 0 otherwise

$$F_x + (n + 1 - L)\delta \geq n + 1 \quad (3)$$

$$F_x + (U - L)\delta \leq n + U - L \quad (4)$$

$\delta = 1$ if $F_x = n$, 0 otherwise

In the two special cases where $n = L$ or $n = U$, it is equivalent and simpler to model the expressions $F_x \leq n$ or $F_x \geq n$ respectively, rather than $F_x = n$. If neither of these is the case then we may enforce the condition in three steps as follows:

(i) $\delta_1 = 1$ if $F_x \geq n$, 0 otherwise

$$F_x - (U - n + 1)\delta_1 \leq n - 1 \tag{5}$$

$$F_x - (n - L)\delta_1 \geq L \tag{6}$$

(ii) $\delta_2 = 1$ if $F_x \leq n$, 0 otherwise

$$F_x + (n + 1 - L)\delta_2 \geq n + 1 \tag{7}$$

$$F_x + (U - L)\delta_2 \leq n + U - L \tag{8}$$

(iii) $\delta = 1$ if $F_x \leq n \wedge F_x \geq n$, 0 otherwise

An equivalent statement is $\delta = 1$ if $\delta_1 + \delta_2 = 2$, 0 otherwise. Note that at least one of conditions (i) and (ii) must hold, therefore $\delta_1 + \delta_2 \geq 1$. Hence, the single constraint

$$\delta = \delta_1 + \delta_2 - 1 \tag{9}$$

is sufficient.

$\delta = 1$ if $F_x \neq n$, 0 otherwise

Constraints (5) to (8) may be applied and constraint (9) is replaced by

$$\delta = 2 - \delta_1 - \delta_2 \tag{10}$$

3.2 Logical constraints

The use of indicator variables in conjuction with propositions as shown above may be extended to enforce relationships between propostions.

We define indicator variables δ_1 such that $\delta_1 = 1$ if proposition X_i is true and 0 if X_i is false. The following equivalencies taken from Williams (1999) will be prove useful.

$$X_1 \wedge X_2 \text{ is equivalent to } \delta_1 + \delta_2 = 2$$

$$X_1 \vee X_2 \text{ is equivalent to } \delta_1 + \delta_2 \geq 1$$

$$\sim X_1 \text{ is equivalent to } \delta_1 = 0$$

$$X_1 \rightarrow X_2 \text{ is equivalent to } \delta_1 \leq \delta_2$$

$$X_1 \Leftrightarrow X_2 \text{ is equivalent to } \delta_1 = \delta_2$$

$$X_1 \leftrightarrow\sim X_2 \text{ is equivalent to } \delta_1 = 1 - \delta_2$$

3.3 Objective functions

The aim of the puzzles is to find a solution where all the statements are consistent. In most cases we may therefore choose any objective function.

4 MODELS

4.1 Werewolves II

Define variables $x_i = 1$ if person i is a knight and 0 if a knave and $y_i = 1$ if person i is a werewolf and 0 otherwise for $i = 1..3$.

As stated above we choose an arbitrary objective function, for example

$$Maximize\ \ x_1$$

Subject to the stated conditions modeled as follows.

Only one person is a werewolf

$$y_1 + y_2 + y_3 = 1$$

If the statement made by A is true then A is a knight. More formally

$$y_1 = 1 \leftrightarrow x_1 = 1$$

and this is modeled quite simply by

$$y_1 = x_1$$

Similarly, if the statement made by B is true then B is a knight is represented by

$$y_2 = x_2$$

If the statement made by C is true then C is a knight or

$$x_1 + x_2 + x_3 \leq 1 \leftrightarrow x_3 = 1$$

and this may be modeled using constraints (3) and (4) and substituting $F_x = x_1 + x_2 + x_3$, $\delta = x_3$, $n = 1, U = 3$ and $L = 0$ as follows

$$x_1 + x_2 + 3x_3 \geq 2$$

$$x_1 + x_2 + 4x_3 \leq 4$$

An Excel spreadsheet to solve the puzzle is here and an Xpress-Mosel model is here.

4.2 Werewolves IV

If the statement by A is true then A is a knight. More formally,

$$x_1 + x_2 + x_3 \leq 2 \leftrightarrow x_1 = 1$$

and this may be modeled using constraints (3) and (4) and substituting $F_x = x_1 + x_2 + x_3$, $\delta = x_1$, $n = 2, U = 3$ and $L = 0$ as follows

$$4x_1 + x_2 + x_3 \geq 3$$

$$4x_1 + x_2 + x_3 \leq 5$$

If the statement by B is true then B is a knight, or

$$x_3 = 1 \leftrightarrow x_2 = 1$$

which is modeled by

$$x_3 = x_2$$

Only one person is a werewolf

$$y_1 + y_2 + y_3 = 1$$

The werewolf is a knight

$$x_i \geq y_i \text{ for } i = 1..3$$

An Excel spreadsheet to solve the puzzle is here and an Xpress-Mosel model is here.

4.3 The Second Trial

Define subscripts $i = 1..2$ for doors and $j = 1..2$ for prizes (1-lady, 2-tiger) and variables as follows:

$x_{i,j} = 1$ if door i hides prize j, 0 otherwise

$t_i = 1$ if statement on door i is true, 0 otherwise

Any objective function

$$Max\ \ x_{1,1}$$

Each door hides one prize

$$\sum_{j=1}^{2} x_{i,j} = 1 \quad for\ \ i = 1..2$$

The logical condition we wish to model for door 1 is

$$t_1 = 1 \leftrightarrow x_{1,1} + x_{2,1} \geq 1$$

and the constraints to enforce this condition are

$$x_{1,1} + x_{2,1} - 2t_1 \leq 0$$

$$x_{1,1} + x_{2,1} - t_1 \geq 0$$

The condition implied by the statement on door 2 is

$$t_2 = 1 \leftrightarrow x_{1,2} = 1$$

and necessary constraint is

$$t_2 = x_{1,2}$$

In addition, we must constrain that the two statements are either both true or both false as follows.

$$t_1 = t_2$$

An Excel spreadsheet to solve the puzzle is here and an Xpress-Mosel model together with a brief explanation of output is here.

4.4 A Logical Labyrinth

We will now apply the above modeling structures to the rather more difficult Logical Labyrinth puzzle from Section 2.2.2.

Define subscripts $i = 1..9$ and $j = 1..3$ (1-lady, 2-tiger, 3-empty) and as above variables are

$x_{i,j} = 1$ if door i hides prize j, 0 otherwise

$t_i = 1$ if statement on door i is true, 0 otherwise

We will commence by using the following arbitary objective function

$$Max\ \ x_{1,1}$$

We now list the statements from the nine doors and state the relationship between the truth or falsity of each statement and the appropriate t_i variable. Linear constraints are developed in each case to enforce these relationships.

Door 1 - the lady is an odd-numbered room.

$$t_1 = 1 \leftrightarrow x_{1,1} + x_{3,1} + x_{5,1} + x_{7,1} + x_{9,1} = 1$$

This may be enforced by

$$t_1 = x_{1,1} + x_{3,1} + x_{5,1} + x_{7,1} + x_{9,1}$$

Door 2 - This room is empty.

$$t_2 = 1 \leftrightarrow x_{2,3} = 1$$

enforced by

$$t_2 = x_{2,3}$$

Door 3 - Either sign 5 is right or sign 7 is wrong.

$$t_3 = 1 \leftrightarrow t_5 + x_{1,1} \geq 1$$

enforced by

$$t_5 + x_{1,1} - 2t_3 \leq 0$$

$$t_5 + x_{1,1} - t_3 \geq 0$$

Door 4 - Sign 1 is wrong.

$$t_4 = 1 \leftrightarrow t_1 = 0$$

enforced by

$$t_4 = 1 - t_1$$

Door 5 - Either sign 2 or sign 4 is right.

$$t_5 = 1 \leftrightarrow t_2 + t_4 \geq 1$$

enforced by

$$t_2 + t_4 - 2t_5 \leq 0$$

$$t_2 + t_4 - t_5 \geq 0$$

Door 6 - Sign 3 is wrong.

$$t_6 = 1 \leftrightarrow t_3 = 0$$

$$t_6 = 1 - t_3$$

Door 7 - The lady is not in room 1.

$$t_7 = 1 \leftrightarrow x_{1,1} = 0$$

enforced by

$$t_7 = 1 - t_{11}$$

Door 8 - This room contains a tiger and room 9 is empty.

$$t_8 = 1 \leftrightarrow x_{8,2} + x_{9,3} \geq 2$$

enforced by

$$x_{8,2} + x_{9,3} - 2t_8 \leq 1$$

$$x_{8,2} + x_{9,3} - 2t_8 \geq 0$$

Door 9 - This room contains a tiger and room 9 is empty

$$t_9 = 1 \leftrightarrow x_{9,2} + t_3 \geq 2$$

enforced by

$$x_{9,2} + t_3 - 2t_9 \leq 1$$

$$x_{9,2} + t_3 - 2t_9 \geq 0$$

Further conditions of the puzzle are modeled as follows.

Each door hides one prize

$$\sum_{j=1}^{3} x_{i,j} \quad for\ i = 1..9$$

Only one room contains a lady.

$$\sum_{i=1}^{9} x_{i,j} = 1$$

The sign on the lady's door is true.

$$t_i \geq x_{i,1} \text{ for } i = 1..9$$

The sign on the tiger's doors are false.

$$t_i \leq 1 - x_{i,2} \text{ for } i = 1..9$$

An Excel spreadsheet to solve the puzzle is here and an Xpress-Mosel model is here.

Experimentation with the model reveals that if the prisoner had been told that room eight was empty he could not have identified the location of the lady. That is, if $x_{8,3}$ is forced to 1 there is no single feasible solution. He must therefore have been informed that room eight was not empty. This additional feature requires the constraint

$$x_{8,3} = 0$$

and the revised model uniquely identifies the whereabouts of the lady.

5 CONCLUSION

Our experience indicates that the challenge to solve increasingly difficult puzzles provides students with sufficient motivation to master the logical modeling techniques and the drudgery normally associated with drill exercises is scarcely noticed.

Finally, an added educational value in drawing parallels between diverse problem situations is that it may lead the students to infer the need for thinking laterally in the search for solutions to the myriad of complex real world business problems.

6 REFERENCES

Chlond M. J. and C.M. Toase (2002), "IP Modeling of Chessboard Placements and Related Puzzles," *INFORMS Transactions on Education*, Vol. 2, No. 2, http://ite.pubs.informs.org/Vol2No2/ChlondToase/

Smullyan, R. (1978), *What is the Name of this Book?*, Prentice-Hall, NJ.

Smullyan, R. (1979), *The Chess Mysteries of Sherlock Homes*, Alfred A. Knopf, Inc., New York, NY.

Smullyan, R. (1981), *The Chess Mysteries of the Arabian Knights*, Alfred A. Knopf, Inc., New York, NY.

Smullyan, R. (1982), *The Lady or The Tiger*, Alfred A. Knopf, Inc., New York, NY.

Smullyan, R. (1982), *Alice in Puzzleland*, William Morrow and Company Inc., New York, NY.

Smullyan, R. (1985), *To Mock a Mockingbird*, Alfred A. Knopf, Inc., New York, NY.

Smullyan, R. (1987), *Forever Undecided: A Puzzle Guide to Godel*, Alfred A. Knopf, Inc., New York, NY.

Smullyan, R. (1992), *Satan, Cantor and Infinity*, Alfred A. Knopf, Inc., New York, NY.

Smullyan, R. (1997), *The Riddle of Scheherezade*, Alfred A. Knopf, Inc., New York, NY.

Williams H.P. (1999),*Model Building in Mathematical Programming*, John Wiley & Sons, New York, NY.

7 APPENDICES

Werewolves II - Xpress-Mosel model

```
model 'were2'
! Description  : Werewolves II
! Source       : Smullyan, R., (1978),
!                What is the Name of this Book?, Prentice-Hall
! Date written : 20/12/99
! Written by   : M J Chlond

  uses 'mmxprs'

  parameters
    person = 3
  end-parameters

  declarations
    x: array(1..person) of mpvar ! x(i) = 1 if person i is a knight,
                                 ! 0 if a knave
    y: array(1..person) of mpvar ! y(i) = 1 if person i is a werewolf,
                                 ! 0 otherwise
  end-declarations

  any:= x(1)

  ! only one is a werewolf
  pca:= sum(i in 1..person) y(i) = 1

  ! if statement 1 is true then set x(1) = 1, else 0
  lca1:= y(1)-9*x(1) <= 0
  lca2:= y(1)-x(1) >= 0

  ! if statement 2 is true then set x(2) = 1, else 0
  lcb1:= y(2)-9*x(2) <= 0
  lcb2:= y(2)-x(2) >= 0

  ! if statement 3 is true then set x(3) = 1, else 0
  lcc1:= sum(i in 1..person) x(i)+9*x(3) >= 2
  lcc2:= sum(i in 1..person) x(i)+9*x(3) <= 10

  forall(i in 1..person) do
    x(i) is_binary
    y(i) is_binary
  end-do
  minimise(any)
  ! display results
  forall(i in 1..person) do
    write(getsol(x(i)),' ',getsol(y(i)))
    writeln
```

```
  end-do

end-model
```

Werewolves IV - Xpress-Mosel model

```
model 'were4'
! Description  : Werewolves IV
! Source       : Smullyan, R., (1978),
!                What is the Name of this Book?, Prentice-Hall
! Date written : 20/12/99
! Written by   : M J Chlond

  uses 'mmxprs'

  parameters
    person = 3
  end-parameters

  declarations
    x: array(1..person) of mpvar ! x(i) = 1 if person i is a knight,
                                 ! 0 if a knave
    y: array(1..person) of mpvar ! y(i) = 1 if person i is a werewolf,
                                 ! 0 otherwise
  end-declarations

  any:= x(1)

  ! if statement 1 is true then x(1) = 1, else 0
  lca1:= sum(i in 1..person) x(i)+3*x(1) >= 3
  lca2:= sum(i in 1..person) x(i)+3*x(1) <= 5

  ! if statement 2 is true then x(2) = 1, else 0
  lcb:= x(3) = x(2)

  ! only one is a werewolf
  pca:= sum(i in 1..person) y(i) = 1

  ! werewolf is a knight
  forall(i in 1..person)
    pcb(i):= x(i) >= y(i)

  forall(i in 1..person) do
    x(i) is_binary
    y(i) is_binary
  end-do

  minimise(any)

  ! display results
```

```
  forall(i in 1..person) do
    write(getsol(x(i)),' ',getsol(y(i)))
    writeln
  end-do

end-model
```

The Second Trial - Xpress-Mosel model

```
model 'trial2'

! Description  : The Second Trial
! Source       : Smullyan, R., (1991), The Lady or The Tiger, OUP
! Date written : 6/12/99
! Written by   : M J Chlond

  uses 'mmxprs'

  parameters
    door  = 2
    prize = 2 ! 1 = Lady, 2 = Tiger
  end-parameters

  declarations
    x: array(1..door,1..prize) of mpvar ! x(i,j) = 1 if door i hides prize j, else 0
    t: array(1..door) of mpvar  ! t(i) = 1 if statement on door i is true, else 0
  end-declarations

  any:= x(1,1)

  ! each door hides 1 prize
  forall(i in 1..door)
    pca(i):= sum(j in 1..prize) x(i,j) = 1

  ! if statement on door 1 is true then t(1) = 1, else t(1) = 0
    lca1:= x(1,1)+x(2,1)-2*t(1) <= 0
    lca2:= x(1,1)+x(2,1)-t(1) >= 0

  ! if statement on door 2 is true then t(2) = 1, else t(2) = 0
    lcb:= t(2) = x(1,2)

  ! statements either both true or both false
    lcc:= t(1) = t(2)

  forall(i in 1..door,j in 1..prize)
    x(i,j) is_binary
  forall(i in 1..door)
    t(i) is_binary

  minimise(any)
```

```
  ! display results
  write('x =')
  writeln
  forall(i in 1..door) do
    forall(j in 1..prize) do
      write(getsol(x(i,j)),' ')
    end-do
    writeln
  end-do

  writeln
  write('t =')
  writeln

  forall(i in 1..door) do
    write(getsol(t(i)),' ')
    writeln
  end-do

end-model
```

The Logical Labyrinth - Xpress-Mosel model

```
model 'trial12'
! Description  : The Logical Labyrinth
! Source       : Smullyan, R., (1991),
!                The Lady or The Tiger, Oxford University Press
! Date written : 21/12/99
! Written by   : M J Chlond

  uses 'mmxprs'

  parameters
    door  = 9
    prize = 3 ! 1 = Lady, 2 = Tiger, 3 = Empty
  end-parameters

  declarations
    x: array(1..door,1..prize) of mpvar ! x(i,j) = 1 if door
                                        ! i hides prize j, else 0
    t: array(1..door) of mpvar   ! t(i) = 1 if statement
                                ! on door i is true, else 0
  end-declarations

  any:= x(1,1)

  ! if statement on door 1 is true
  ! (i.e. x(1,1)+x(3,1)+x(5,1)+x(7,1)+x(9,1) = 1 )
  ! then t(1) = 1, else t(1) = 0
```

```
lca:= t(1) = x(1,1)+x(3,1)+x(5,1)+x(7,1)+x(9,1)

! if statement on door 2 is true
! (i.e. x(2,3)=1) then t(2) = 1, else t(2) = 0
lcb:= t(2) = x(2,3)

! if statement on door 3 is true
! (i.e. t(5)+x(1,1) > 1 ) then t(3) = 1, else t(3) = 0
lcc1:= t(5)+x(1,1)-2*t(3) <= 0
lcc2:= t(5)+x(1,1)-t(3) >= 0

! if statement on door 4 is true
! (i.e. t(1) = 0) then t(4) = 1, else t(4) = 0
lcd:= t(4) = 1-t(1)

! if statement on door 5 is true
! (i.e. t(2)+t(4) > 1) then t(5) = 1, else t(5) = 0
lce1:= t(2)+t(4)-2*t(5) <= 0
lce2:= t(2)+t(4)-t(5) >= 0

! if statement on door 6 is true
! (i.e. t(3) = 0 ) then t(6) = 1, else t(6) = 0
lcf:= t(6) = 1-t(3)

! if statement on door 7 is true
! (i.e. x(1,1) = 0) then t(7) = 1, else t(7) = 0
lcg:= t(7) = 1-x(1,1)

! if statement on door 8 is true
! (i.e. x(8,2)+x(9,3) = 2 ) then t(8) = 1, else t(8) = 0
lch1:= x(8,2)+x(9,3)-2*t(8) <= 1
lch2:= x(8,2)+x(9,3)-2*t(8) >= 0

! if statement on door 9 is true
! (i.e. x(9,2)+t(3) = 2) then t(9) = 1, else t(9) = 0
lci1:= x(9,2)+t(3)-2*t(9) <= 1
lci2:= x(9,2)+t(3)-2*t(9) >= 0

! each door hides 1 prize
forall(i in 1..door)
  pca(i):= sum(j in 1..prize) x(i,j) = 1

! only one room contains lady
pcb:= sum(i in 1..door) x(i,1) = 1

! sign on lady’s door is true
forall(i in 1..door)
  lck(i):= t(i) >= x(i,1)
```

```
  ! sign on tigers' doors are false
  forall(i in 1..door)
    lcl(i):= t(i) <= 1 - x(i,2)


  ! if room 8 is empty then not enough information to pinpoint lady
  ! min and max x(7,1) give different results

  ! room 8 is empty
  !pcc:= x(8,3) = 1

  ! if room 8 is not empty then enough information
  ! min and max x(7,1) gives same results
  ! if the prisoner was able to deduce where the lady was then
  ! room 8 must not have been empty

  ! room 8 is not empty
  pcc:= x(8,3) = 0

  forall(i in 1..door,j in 1..prize)
    x(i,j) is_binary
  forall(i in 1..door)
    t(i) is_binary

  minimise(any)

  ! display results

  write('x =')
  writeln

  forall(i in 1..door) do
    forall(j in 1..prize) do
      write(getsol(x(i,j)),' ')
    end-do
    writeln
  end-do

  writeln
  write('t =')
  writeln

  forall(i in 1..door) do
    write(getsol(t(i)),' ')
    writeln
  end-do

end-model
```