

Challenge June-2021

Where is gold?

A solution with DT5GL by Jack Jansonius – 20 June 2021

Personally, I am more interested in solving real world problems, such as an organ donation or Covid19 testing than solving logic puzzles, but for this challenge I am quite pleased to make an exception.

First of all, because of the great, fast initial contribution by Jan Vanthienen; it may be revealing that his first simple solution (Starting from the boxes) particularly appeals to me and his second solution (Starting from the statements) does not appeal to me at all. But distinguishing between these approaches is of course excellent!

Second, because although I had an interactive solution to his first solution very quickly, I did not have a fully automated one. Put simply, my tool had a condition "box in [1,2,3]" but not a condition "next box in [1,2,3]", that is, the ability to have the variable box go through the values 1, 2, 3 sequentially.

For this reason, I will first provide the interactive version of the solution, where the box number to be examined will be prompted to the user.

With a slight modification of this interactive version, the box numbers to be examined will then be automated.

But first, here is the description of the challenge as shown on the website:

Challenge June-2021

Where is gold?



There are three boxes, but only one of them has gold inside. Additionally, each box has a message printed on it. One of these messages is true and the other two are lies.

The first box says, "Gold is in this box".

The second box says, "Gold is not in this box".

The third box says, "Gold is in not in Box 1".

Which box contains the gold?

Implementation of the decision tables in DT5GL; interactive version:

Table 0:

If:	0 1 2 3
'Next box'	Y Y Y N
Box in [1,2,3]	Y Y N -
TruthSum = 1	Y N - -
Then:	
Result is Finished	X
Result is Wrong_Boxnr	X
Result is Gold_in_box	X
Result is Gold_not_in_box	X
#	

Proposition: 'Next box'
Askable_using: "Next box?"

Attribute: Box
Askable_using: "Box nr?"

rTable 1:	
If:	0 1 2
Box = 1	Y N N
Box = 2	- Y N
Box = 3	- - Y
Then:	
Box1_here = 1	X
Box2_not_here = 1	X X
Box3_not_box1 = 1	X X
#	

Attribute: TruthSum
Equals: Box1_here + Box2_not_here + Box3_not_box1

Attribute: Truthspeaker Type: Integer
Equals: 1 if Box1_here else 2 if Box2_not_here else 3 if Box3_not_box1 else 0

GoalAttribute: Result
Repeat_until: Finished

Case: Finished
Print: "Finished!"
Case: Wrong_Boxnr
Print: "Enter 1, 2 or 3!"

Case: Gold_in_box
Print: "Gold in box nr %s!" Box
Print: "Truthspeaker is box: %s." Truthspeaker
Print: "-----"
Case: Gold_not_in_box
Print: "Gold not in boxnr %s!" Box
Print: "-----"

Some test cases for the interactive version:

```
"Next box? (y/n)? > y
"Box nr?"
(int)> 1
```

```
Gold not in boxnr 1!
-----
"Next box? (y/n)? > y
"Box nr?"
(int)> 2
```

```
Gold in box nr 2!
Truthspeaker is box: 3.
-----
"Next box? (y/n)? > y
"Box nr?"
(int)> 3
```

```
Gold not in boxnr 3!
-----
"Next box? (y/n)? > y
"Box nr?"
(int)> 4
```

```
Enter 1, 2 or 3!
"Next box? (y/n)? > n
Finished!
```

Implementation of the decision tables in DT5GL; automated version:

Table 0:

If:	0 1 2
Next Box in [1,2,3]	Y Y N
TruthSum = 1	Y N -
Then:	
Result is Finished	X
Result is Gold_in_box	X
Result is Gold_not_in_box	X
#	

rTable 1:

If:	0 1 2
Box = 1	Y N N
Box = 2	- Y N
Box = 3	- - Y
Then:	
Box1_here = 1	X
Box2_not_here = 1	X X
Box3_not_box1 = 1	X X
#	

Attribute: TruthSum

Equals: Box1_here + Box2_not_here + Box3_not_box1

Attribute: Truthspeaker Type: Integer

Equals: 1 if Box1_here else 2 if Box2_not_here else 3 if Box3_not_box1 else 0

GoalAttribute: Result

Repeat_until: Finished

Case: Finished

Print: "Finished!"

Case: Gold_in_box

Print: "Gold in box nr %s!" Box

Print: "Truthspeaker is box: %s." Truthspeaker

Print: "-----"

Case: Gold_not_in_box

Print: "Gold not in boxnr %s." Box

Print: "-----"

Test run:

Gold not in boxnr 1.

Gold in box nr 2!

Truthspeaker is box: 3.

Gold not in boxnr 3.

Finished!

Some minor changes to the automated version.

Suppressing the "Gold not in box" notification can be done easily with:

```
Case: Gold_not_in_box
Print: "#REM# Gold not in boxnr %s."      Box
```

and results in:

```
Gold in box nr 2!
Truthspeaker is box: 3.
-----
Finished!
```

Stopping when the gold is found is easy with:

```
GoalAttribute: Result
Repeat_until: Gold_in_box, Finished
```

and results in:

```
Gold not in boxnr 1.
-----
Gold in box nr 2!
Truthspeaker is box: 3.
-----
```

Examining some additional boxes can be done with:

```
Next Box in [1,2,3,4,5]
```

and has as a result:

```
Gold not in boxnr 1.
-----
Gold in box nr 2!
Truthspeaker is box: 3.
-----
Gold not in boxnr 3.
-----
Gold not in boxnr 4.
-----
Gold not in boxnr 5.
-----
Finished!
```