# A NOTE ON KNIGHTS, KNAVES, AND TRUTH TABLES

H. REINHART and J. BOYD
*St. Christopher's School*
*Richmond, Virginia 23226*

**Abstract.**

Student and teacher (in the order above) have taken on a project of trying to understand something about Boolean algebra, logic circuits, and applications with the aid of *Mathematica*. We quickly recognized that the logical puzzles popularized by Raymond Smullyan that involve Knights (truth tellers) and Knaves (liars) are ideally suited for analysis by Boolean methods and truth tables. with a big boost from *Mathematica*. Not only is there a lot of mathematics to be learned, there is a great deal of fun to be had. The topic seems to us to be an ideal vehicle for exposing young high school and undergraduate college students to wonderful mathematics outside of the standard Advanced Placement Calculus stream.

## Introduction

Most people who learn, teach, or do mathematics are familiar with Raymond Smullyan's mind-bending puzzles and, in particular, with the logical problems inherent in the conversations of knights and knaves. A knight tells the truth under all circumstances while a knave always lies. Now suppose that every inhabitant of a certain island is either a knight or a knave. To be precise, as one needs to be in Smullyan's world, the preceding "or" is used in the exclusive sense. An ordinary visitor to the island from the mainland would be unable to judge whether an islander is a knight or a knave just from his appearance or from what he says.

Suppose that our visitor meets islanders A, B, and C and that

A says that either B or C is a knight;
B says that A is a knave; and,
C says that A says that B is a knave.

The visitor's problem and our problem, too is to determine whether or not there exists any combination of A, B, and C as knights or knaves, such that their linked comments become logically consistent. In the note to follow, we use a little binary Boolean algebra and construct truth tables to discover which (if any) combinations of knights and knaves will lead to logical consistency when all of their comments are taken together. We have intentionally avoided writing "A's, B's and C's statements" since the statements that we shall test through the construction of truth tables are not the comments actually made by the islanders, but statements derived from their comments.

**The Statements**

First, we observe that it might seem that the pairings "logically consistent or not consistent" and "valid or not valid" make more sense than "true or false" in the context of knights and knaves. However, since "true or false" is firmly embedded in the vocabulary of Boolean algebra, we shall ask for combinations of knights and knaves that make their linked statements true.

Let us denote A's comment by CA. Then, the associated statement to be tested is

$$SA = (CA) \text{ exclusive or } (\text{not } A) \tag{1}$$

where the statement A is that A is a knight and not A is the statement that A is a knave. The use of the Boolean exor function in deriving the statements from the comments of A, B, and C makes great sense since Xor[P, Q] is true if and only if the truth values of P and Q are different. When we use *Mathematica* to construct our truth tables, we must rewrite equation 1 as

$$SA = \mathrm{Xor}[CA, \neg A].$$

Then the three statements derived from the listing of comments by A, B, and C as given above are

$$SA = \mathrm{Xor}[B \vee C, \neg A],\ SB = \mathrm{Xor}[\neg A, \neg B], \text{ and } SC = \mathrm{Xor}[\mathrm{Xor}[\neg B, \neg A], \neg C]. \tag{2}$$

The three statements taken together as a single statement will be true if and only if all three individual statements are true. Thus, we are led to the logical conjunction to be implemented with *Mathematica* and its “And” function. Thus, our test for the truth of the islanders' three statements will be determined by evaluating the Boolean function

$$F[A, B, C] = \mathrm{And}[SA, SB, SC] = SA \wedge SB \wedge SC. \tag{3}$$

If and only if F[A, B, C] takes the value "true" do we say that the information put out by A, B, and C together is true or consistent.

**Truth Tables**

We have found great value in constructing the knights' and knaves' tables by hand. Just as students in high school who have refrained from using their calculators too soon but who have practiced long multiplications and divisions by hand will know their number facts better and understand polynomials more quickly than those who have not, so those who have filled in their own truth tables and not turned straight to *Mathematica* will have gained a real feeling for how truth tables work. However, the purpose of our note is to explain how one can solve a class of difficult logical problems by method rather than by resort to ingenuity. So for convenience, we turn quickly to *Mathematica* to display the truth table that will test the statements for A, B, and C as given above. The column headings for the table are those defined by equations 1, 2, and 3.

**Table 1.**
**Truth Values for the Introductory Problem.**

```
SA = Xor[(B ∨ C), ¬A];
SB = Xor[¬A, ¬B];

SC = Xor[Xor[¬B, ¬A], ¬C];

TableForm[BooleanTable[{A, B, C, SA, SB, SC, SA ∧ SB ∧ SC}],

TableHeadings → {None, {"A", "B", "C", "SA", "SB", "SC", "SA ∧ SB ∧ SC"}}]
```

| "A" | "B" | "C" | "SA" | "SB" | "SC" | "SA ∧ SB ∧ SC" |
|---|---|---|---|---|---|---|
| True | True | True | True | False | False | False |
| True | True | False | True | False | True | False |
| True | False | True | True | True | True | True |
| True | False | False | False | True | False | False |
| False | True | True | False | True | True | False |
| False | True | False | False | True | False | False |
| False | False | True | False | False | False | False |
| False | False | False | True | False | True | False |

We see that if and only if A and C are knights and B is a knave, their statements are consistent.

After an aside concerning the liar's paradox, the remainder of our note is devoted to the stating and solution of four more problems. Our intention is to emphasize that method may win the prize even when ingenuity and deep insight may fail.

**The Liar's Paradox**

Since we are concerned with knights who always tell the truth and knaves who always lie, it would seem instructive to consider the liar's paradox. We may take the statement

A says that A is a knave

as the paradox. The Boolean representation of the paradox is Xor[¬A, ¬A] and here is its truth table.

**Table 2.**
**The Liar's Paradox.**

```
SA = Xor[¬A, ¬A];
TableForm[BooleanTable[{A, ¬A, SA}], TableHeadings → {None, {"A", "Not A", "SA"}}]
```

| "A" | "Not A" | "SA" |
|---|---|---|
| True | False | False |
| False | True | False |

We see that our solution scheme quickly disposes of a logical conundrum. Next, we present our four problems which feature in succession two, three, four, and five islanders having their say. We hope that our readers will agree that, with the help of a little Boolean algebra, we can solve puzzles too complicated for one to solve by conversational methods alone.

<u>Problem 1</u>. Our characters are islanders A and B.

A says B says A is a knave, and
B says A is a knight.

We note that since A says that B says . . . , we must construct SA with a composition of Xor's.

The statements to be tested are SA = Xor[Xor[¬A, ¬B], ¬A] and SB = Xor[A, ¬B]. Again, we turn to *Mathematica* and obtain the results shown in Table 3.

**Table 3.**
**Truth Values for Problem 1**

```
SA = Xor[Xor[¬A, ¬B], ¬A];
SB = Xor[A, ¬B];
tbl = BooleanTable[{A, B, ¬A, ¬B, SA, SB, SA ∧ SB}];
TableForm[tbl, TableHeadings → {None, {"A", "B", "¬A", "¬B", "SA", "SB", "SA ∧ SB"}}]
```

| "A" | "B" | "¬A" | "¬B" | "SA" | "SB" | "SA ∧ SB" |
|---|---|---|---|---|---|---|
| True | True | False | False | False | True | False |
| True | False | False | True | True | False | False |
| False | True | True | False | False | False | False |
| False | False | True | True | True | True | True |

We see that the two statements SA and SB are consistent if and only if both A and B are knaves.

Problem 2.

A says C is a knight;

B says A is a knight; and,

C says B is a knight or A is a knave.

Our statements are SA = Xor[C, ¬A], SB = Xor[A, ¬B], and Xor[(B | | ¬A), ¬C]. Here is our truth table.

**Table 4.**

**Truth Values for Problem 2.**

```
SA = Xor[C, ¬A];
SB = Xor[A, ¬B];
SC = Xor[(B ∨ ¬A), ¬C];
tbl = BooleanTable[{A, B, C, SA, SB, SC, SA ∧ SB ∧ SC}];
TableForm[tbl, TableHeadings → {None, {"A", "B", "C", "SA", "SB", "SC", "SA ∧ SB ∧ SC"}}]
```

| "A" | "B" | "C" | "SA" | "SB" | "SC" | "SA ∧ SB ∧ SC" |
|---|---|---|---|---|---|---|
| True | True | True | True | True | True | True |
| True | True | False | False | True | False | False |
| True | False | True | True | False | False | False |
| True | False | False | False | False | True | False |
| False | True | True | False | False | True | False |
| False | True | False | True | False | False | False |
| False | False | True | False | True | True | False |
| False | False | False | True | True | False | False |

We see that the three statements SA, SB, and SC are consistent if and only if all three islanders are knights.

We also observe that the single change of switching A's comment to "C is a knave " drives the solution to the new state in which A and B are knaves and C is a knight. This new result is displayed in Table 5 below.

**Table 5.**

**Truth Values for the Changed Problem 2**

```
SA = Xor[¬C, ¬A];

SB = Xor[A, ¬B];
SC = Xor[(B ∨ ¬A), ¬C];
tbl = BooleanTable[{A, B, C, SA, SB, SC, SA ∧ SB ∧ SC}];
TableForm[tbl, TableHeadings → {None, {"A", "B", "C", "SA", "SB", "SC", "SA ∧ SB ∧ SC"}}]
```

| A | B | C | SA | SB | SC | SA ∧ SB ∧ SC |
|---|---|---|---|---|---|---|
| True | True | True | False | True | True | False |
| True | True | False | True | True | False | False |
| True | False | True | False | False | False | False |
| True | False | False | True | False | True | False |
| False | True | True | True | False | True | False |
| False | True | False | False | False | False | False |
| False | False | True | True | True | True | True |
| False | False | False | False | True | False | False |

We suspect that interesting results might follow from recording the effects of successive small changes in the statements SA, SB, and SC.

Problem 3.

A says B says C is a knave;

B says A is a knave or C is a knight;

C says both B and D are knights; and,

D says B is a knight.

Our statements are SA= Xor[Xor[¬C, ¬B], ¬A], SB = Xor[(¬A ∨ C), ¬B], SC = Xor[(B ∧ D), ¬C], and SD = Xor[B, ¬D]. Since our table contains sixteen rows, we save space by printing only the row (or rows) in which SA ∧ SB ∧ SC ∧ SD = true.

**Table 6.**

**The Solution for Problem 3.**

```
SA = Xor[Xor[¬C, ¬B], ¬A];
SB = Xor[(¬A ∨ C), ¬B];
SC = Xor[(B ∧ D), ¬C];
SD = Xor[B, ¬D];
tbl = BooleanTable[{A, B, C, D, SA, SB, SC, SD, SA ∧ SB ∧ SC ∧ SD}];
TableForm[tbl,
  TableHeadings →
   {None, {"A", "B", "C", "D", "SA", "SB", "SC", "SD", "SA ∧ SB ∧ SC ∧ SD"}}];
```

```
Do[If[tbl[[k, 9]] == True, Print["Row #", k, ": ", tbl[[k]]]], {k, 1,16}]
Row #9: {False, True, True, True, True, True, True, True, True}
```

We see that consistency is achieved when A is a knave and B, C, and D are all knights. The full truth table can be viewed by removing the semicolon at the end of the next-to-the-last line of the program above.

<u>Problem 4</u>.

A says B says both C and D are knights;

B says only one of A and E are knaves;

C says A says E is a knight;

D says C says B is a knight; and,

E says D says C is a knave.

Our statements are SA= Xor[Xor[(C ∧ D), ¬B], ¬A], SB = Xor[Xor[¬A,¬E],¬B], SC = Xor[Xor[E, ¬A], ¬C], SD = Xor[Xor[B, ¬C], ¬D], and SE = Xor[Xor[¬C ,¬D], ¬E.] Since our large table contains thirty-two rows, we save space by printing only the row (or rows) in which

SA ∧ SB ∧SC ∧ SD ∧ SE = true.

**Table 7.**
**The Solution for Problem 4.**

```
SA = Xor[Xor[(C ∧ D), ¬B], ¬A];
SB = Xor[Xor[¬A, ¬E], ¬B];
SC = Xor[Xor[E, ¬A], ¬C];
SD = Xor[Xor[B, ¬C], ¬D];
```

```
SE = Xor[Xor[¬C, ¬D], ¬E];
tbl = BooleanTable[{A, B, C, D, E, SA, SB, SC, SD, SE, SA ∧ SB ∧ SC ∧ SD ∧ SE}];
TableForm[tbl,
  TableHeadings → {None, {"A", "B", "C", "D", "E", "SA", "SB", "SC", "SD", "SE",
   "SA ∧ SB ∧ SC ∧ SD ∧ SE"}}];

Do[If[tbl[[k, 11]] == True, Print["Row #", k, ": ", tbl[[k]]]], {k, 1,32}]
Row #11: {True, False, True, False, True, True, True, True, True, True, True}
```

We see that what the five islanders say is consistent if and only if A, C, and E are knights, and B and D are knaves. Again, the full truth table can be viewed by removing the semicolon at the end of the next-to-the-last line of the program above.

**Conclusion**

We hope that we have convinced our readers that using *Mathematica* to construct truth tables is both an interesting and efficient way of attacking Smullyan's problems about knights and knaves. Finally, we are indebted to Mr. Frank Kiefer for sparking our interest in knights and knaves.