3.13. Knight-Knave Puzzles: What They Are, and How to Solve Them

We are arrant knaves all: believe none of us.
– William Shakespeare, **Hamlet** iii, 1

1. Solving a Knight-Knave Puzzle. Knight-Knave puzzles provide an additional application of formal logic. Such a puzzle is a set on an island – the Island of Knights and Knaves – where each inhabitant is either a knave or a knave (but not both). **A knight always tell the truth**, and **a knave always lies**. Because knights and knaves are otherwise indistinguishable, communication with them is no easy thing. For instance: we can ask the person which he or she is, but both knights and knaves can equally well respond "I'm a Knight" (knights because it's true, knaves because it's not). So that answer tells us nothing – certainly not the status of the speaker.

Yet it is possible to gather information from the inhabitants of the island, through a clever application of formal logic. An example illustrates how.

Suppose on the beach we meet two inhabitants of the island named "P" and "Q," and **P says** "**Either I'm a knave or Q is a knight**."

The **first step** in sorting out such a situation is to **translate the claim into formal language**. Let's use the following translation key:

P: P is a knight
Q: Q is a knight

And we needn't appeal to further sentence letters in order to claim that someone is a knave. Because each inhabitant is either a knight or a knave – but **not both** – the sentence "**P** is not a knight" is here equivalent (in truth value) to "**P** is knave". So we can apply the following translation shortcut.

~P: P is a knave
~Q: Q is a knave

We can now translate P's claim into formal notation. P said:

Either I'm a knave or Q is a knight.

This translates into the following formal sentence.

$(\sim P \vee Q)$

To this sentence we apply the following two 'laws' of the island.

The **First Law** of the island is: **if the speaker is a knight, then what the speaker said is really so**. **P** said: "$(\sim P \vee Q)$". So by the First Law: **if P is a Knight, then $(\sim P \vee Q)$**. Following our earlier translation key, that conditional translates like this.

$$(1)\ (P \rightarrow (\sim P \vee Q))$$

The **Second Law** of the island is the converse of the first: **if what the speaker says really is so, then the speaker is a Knight**. Applied to **P**'s utterance, that means: **If $(\sim P \vee Q)$, then P is a knight**. That conditional translates as follows.

$$(2)\ ((\sim P \vee Q) \rightarrow P)$$

By the very nature of knights and knaves we know we're in a situation where **both these conditionals are true**. Using semantic methods (truth tables or a truth tree) we can work out the status of **P** and **Q** by tracking down the situation which makes **both conditionals true** ('simultaneously satisfies' them).

For instance, truth tables reveal that there's only one valuation making both conditionals true: the **first** valuation.

P	**Q**	**~P**	**$(\sim P \vee Q)$**	**$(P \rightarrow (\sim P \vee Q))$**	**$((\sim P \vee Q) \rightarrow P)$**
1	1	0	1	**1**	**1**
1	0	0	0	0	1
0	1	1	1	1	0
0	0	1	1	1	0

But the first valuation makes sentences "P" and "Q" true.

P	**Q**	**~P**	**(~P ∨ Q)**	**(P → (~P ∨ Q))**	**((~P ∨ Q) → P)**
1	**1**	0	1	**1**	**1**
1	0	0	0	0	1
0	1	1	1	1	0
0	0	1	1	1	0

By the earlier translation key, that means the sentence "**P** is a knight" is true, and sentence "**Q** is a knight" is true. Hence this is a situation **P and Q are both Knights**.

Answer:
P is a knight.
Q is a knight.

To use a **truth tree** to solve the puzzle, we assume both sentences true, and find the situation where that's possible (the tree path which doesn't close).

2. A Biconditional Shortcut. The above instructions require us to build a conditional and its converse. But since those two sentences together are equivalent to a biconditional, we can save a step in our truth tables by simply building a biconditional sentence of the following sort.

- On the left half of the biconditional, place **the claim that the speaker is a knight**. (For example, if **P** is speaking, use sentence letter "P" as the left half of the biconditional.)
- On the right half of the biconditional, place **the sentence that the speaker said**.

So in the earlier example, where speaker **P** says "(~P ∨ Q)", we construct the following biconditional.

(1) $(\underline{P} \leftrightarrow \underline{(\sim P \vee Q)})$

This biconditional is guaranteed to be true on the Island of Knights and Knaves (since the two conditionals are guaranteed to be true there, and the biconditional will be true whenever those two conditionals are).

The truth table valuation making this biconditional true tells us whether each person mentioned is a knight or a knave.

P	**Q**	**~P**	**(~P ∨ Q)**	**(P ↔ (~P ∨ Q))**
1	1	0	**1**	**1**
1	0	0	0	0
0	1	1	1	1
0	0	1	1	1

Once again, the sentence letters "P" and "Q" are both true – so both **P** and **Q** are knights.

To use a **truth tree** to solve the puzzle, we assume that the biconditional is true (on the left side) and find the situation where that's possible (the tree path which doesn't close).

3. Knight-Knave Puzzles with More than Two People. The above procedure scales up naturally to puzzles involving more than two people. Here again we build a biconditional for each sentence uttered, then use semantic methods to find the situation making all those biconditionals true.

For example, suppose Q says "P and I aren't both knights," and R says "Neither P nor Q is a knight". We construct two biconditionals based on this two sentences, like so.

P: P is a knight
Q: Q is a knight
R: R is a knight

$(Q \leftrightarrow \sim(P \wedge Q))$
$(R \leftrightarrow \sim(P \vee Q))$

We then use truth tables or a truth tree to find the situation(s) where both these sentences are true.

(Note: when solving a three-person puzzle with a truth tree, more than one tree path may stay open, but with all the paths agreeing on the truth value of each sentence letter. For example, there may be two open paths, both of which make "P" and "Q" true, but "R" false.)

Summary:
Solving a "Knight-Knave" Puzzle

- **Translate** what the speaker said into formal language.

- (Following the laws of the island), **build two conditionals** guaranteed to be true.

    1. If that person is a Knight, then [sentence the person said].
    2. If [sentence the person said], then that person is a Knight.

- Use semantic methods (truth tables or a truth tree) to work out **where those two sentences are both true**. That reveals the status (knight or knave) of the individuals in that puzzle.

- As a shortcut: instead of building the two conditional sentences lists above, just build a **biconditional** guaranteed to be true.

    That person is a knight if and only [sentence that person said].

    Then use semantic methods to find where the biconditional is true.

- If there is more than one speaker in a puzzle, build a biconditional for each sentence uttered, then use semantic methods to find the situation where **all the biconditionals are true**.