Conditional Translation, Truth Table, Validity Problem: Discussion

Problem: Use the translation table provided to translate the following English argument into formal language; then build truth tables to decide on the validity of the argument.

> 1. It's not the case that: Jack is a bird who can fly.
> ----------
> (so) If Jack is a bird, then Jack can't fly.
> (**P**: Jack is a bird; **Q**: Jack can fly.)

Discussion: We begin translating by picking out the form phrases in the premise. Here there are two: the negation phrase "it is not the case that," and the relative clause "who can fly," which is treated as a kind of disguised conjunction.

It is not the case that Jack is a bird **who** can fly.

"Jack is a bird who can fly" is treated as a conjunction (equivalent to "Jack is a bird, and Jack can fly")

Jack is a bird [**who** can fly]
$(P \wedge Q)$

"It is not the case that" is outside this conjunction, negating it. So the complete sentence is translated as follows.

It is not the case that Jack is a bird who can fly.
$\sim(P \wedge Q)$

The conclusion has two form phrases: "if… then" and "n't".

If Jack is a bird, **then** Jack ca**n't** fly.

Since “then” is by the comma, we suppose that “if… then is gluing the two parts of the sentences together (with “n’t” buried in the consequent). “If… then” is translated by the arrow (with parentheses).

> **If** Jack is a bird, **then** Jack can’t fly.
> (Jack is a bird $\rightarrow$ Jack can’t fly)

We replace the subject matter sentences with sentence letters, following the translation table.

> $(P \rightarrow$ n’t $Q)$

“N’t” is translated by the tilde.

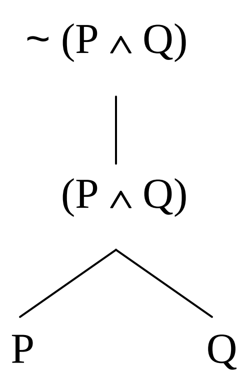
> $(P \rightarrow \sim Q)$

So the whole argument is translated like so.

$\sim(P \wedge Q)$

$\therefore\ (P \rightarrow \sim Q)$

Next we build a truth table for the sentence. As always, a construction tree is a reliable way of seeing which steps are required to build its truth table.



Since the sentence begins with the two sentence letters “P” and “Q,” our truth table will as well.

P	**Q**
1	1
1	0
0	1
0	0

We then add places in the table for the remaining sentences in the tree: “(P ∧ Q)” and “~ (P ∧ Q)”.

P	**Q**	**(P ∧ Q)**	**~ (P ∧ Q)**
1	1		
1	0		
0	1		
0	0		

“(P ∧ Q)” follows the conjunction rule: the sentence is only true when both parts are true – in the first valuation. It’s false in the other valuations.

●	▲	(● ∧ ▲)
1	1	**1**
1	0	0
0	1	0
0	0	0

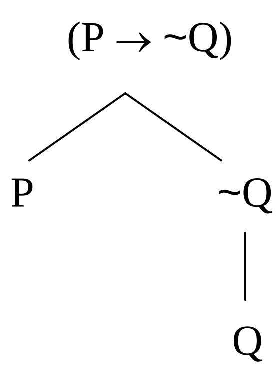
P	**Q**	**(P ∧ Q)**	**~ (P ∧ Q)**
1	1	**1**	
1	0	**0**	
0	1	**0**	
0	0	**0**	

The negation this sentence follows the Negation Rule: where the original sentence was true (Valuation 1), its negation is false. Where the original sentence was false (Valuations 2, 3, and 4), the negation is true.

●	~●
1	0
0	1

P	**Q**	**(P ∧ Q)**	**~ (P ∧ Q)**
1	1	**1**	**0**
1	0	**0**	**1**
0	1	**0**	**1**
0	0	**0**	**1**

The conclusion was built like so.



A truth table for this sentence will require truth tables for all its parts: “P,” “Q,” and “~Q”. The truth tables for “P” and “Q” already appear; so we need only built a truth table for “~Q,” and then for the entire conclusion “(P → ~Q)”.

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

“~Q” follows the Negation Rule: where “Q” is true (Valuations 1 and 2), “~Q” is false; and where “Q” is false (Valuations 3 and 4) “~Q” is true.

●	~●
1	0
0	1

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

Now we have a truth table for the antecedent “P” and for the consequent “~Q” – all the parts needed for a truth table for the whole conditional “(P → ~Q)”. This sentence follows the Conditional Rule: a conditional is only **false** when the **antecedent** “P” is **true**,” but the **consequent** “~Q” is **false**. That’s Valuation 1.

●	▲	(● → ▲)
1	1	1
1	0	0
0	1	1
0	0	1

Antecedent				**Consequent**	
P	**Q**	**(P ∧ Q)**	**~ (P ∧ Q)**	**~Q**	**(P → ~Q)**
1	1	1	0	**0**	**0**
1	0	0	1	1	
0	1	0	1	0	
0	0	0	1	1	

The conditional is true in the other three valuations.

Antecedent					**Consequent**
P	**Q**	**(P ∧ Q)**	**~ (P ∧ Q)**	**~Q**	**(P → ~Q)**
1	1	1	0	0	0
1	0	0	1	1	**1**
0	1	0	1	0	**1**
0	0	0	1	1	**1**

With the truth table for the premise and the conclusion built, we can assess the validity of the argument. We only focus on valuations that make all the premises true – here, all *one* of them.

The premise "~(P ∧ Q)" is true in Valuations 2, 3, and 4.

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

But note: in every one of those three valuations, the conclusion is also true. So: whenever the premises are true, the conclusion is true.

So: the argument is **valid**.