Conditional Translation and Truth Table HW: Discussion

***Problem*:** Use the translation table given to translate the following sentence into formal notation, and then build a truth table for that formal sentence.

> If it's not sunny, then we won't have a picnic.
> (**P**: It's sunny; **Q**: We'll have a picnic.)

Discussion: We begin translating by picking out the form phrases in the sentence. Here there are three: "if… then," "not," and "n't".

> **If** it's **not** sunny**, then** we wo**n't** have a picnic**.**

What's left beyond these form phrases are the two subject matter sentences Following the translation table, we put sentence letters in place of the subject matter sentences – leaving this.

> **If not** P**, then n't** Q

Since the comma (marking a main break) comes right by the word "then," we conclude that "if… then" is the main connective of the sentence. "If…then" is translated by the arrow, and the ordinary conditional phrase "if" comes right before the antecedent. So "not P" must be the antecedent, going before the arrow, and "n't Q" is the consequent, going after the arrow.

> **If not** P**, then n't** Q

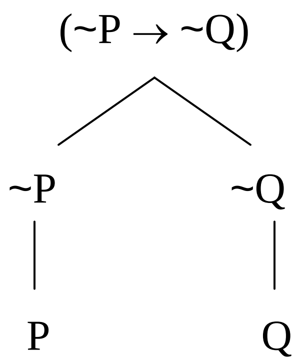
> (**not** P $\rightarrow$ **n't** Q)

Of course both "not" and "n't" are negation phrases, translated by the tilde.

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

This completes the translation.

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" is just the negation of "P," and follows the Negation Rule: when "P" is true, its negation "~P" is false; and when "P" is false, its negation "~P" is true.

●	~●
1	0
0	1

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

Likewise the Negation Rule says that when "Q" is true, its negation "~Q" is false; and when "Q" is false, its negation "~Q" is true.

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

Once we have the truth table for the antecedent, "~P" and the consequent "~Q," we can build the truth table for the whole conditional – using the Conditional Rule, of course.

●	▲	(● → ▲)
1	1	1
1	0	0
0	1	1
0	0	1

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

In the first valuation, the antecedent is false and the consequent is false. In this sort of valuation, the Conditional Rule says the whole conditional is **true**.

●	▲	(● → ▲)
1	1	1
1	0	0
0	1	1
0	**0**	**1**

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

In the second valuation, the antecedent is false, but the consequent is true. This sort of valuation makes a conditional **true**.

●	▲	(● → ▲)
1	1	1
1	0	0
0	**1**	**1**
0	0	1

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

In the third valuation, the antecedent is true, but the consequent is false. The Conditional Rule says a conditional is **false** in this sort of valuation.

	●	▲	(● → ▲)
	1	1	1
⇨	**1**	**0**	**0**
	0	1	1
	0	0	1

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

The last valuation makes both the antecedent and the consequent true. According to the Conditional Rule, a conditional is **true** in this sort of valuation.

	●	▲	(● → ▲)
⇨	**1**	**1**	**1**
	1	0	0
	0	1	1
	0	0	1

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

This completes the assignment.

If it's not sunny, then we won't have a picnic.

P: It's sunny
Q: We will have a picnic

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

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