Truth Table Problem:
Web CT Discussion

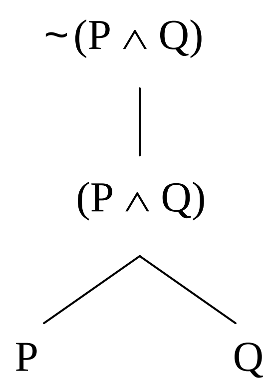
Build truth tables for the following two formal sentences.

(1) $\sim(P \wedge Q)$
(2) $(\sim P \wedge Q)$

Sentence (1): $\sim(P \wedge Q)$

Discussion: This sentence contains both a wedge and a tilde. Since the very first (left-most) symbol is a tilde, that tilde is the *main connective*. So this sentence is a negation. (If the wedge were the main connective, the leftmost symbol would have been a left parenthesis.)

The construction tree for this sentence is as follows.



And the truth table for the sentence reflects this structure.

P	**Q**	**$(P \wedge Q)$**	**$\sim(P \wedge Q)$**

We put a “2” above each sentence letter, and multiple them together, getting 4. We need 4 valuations to go through all the combinations.

4 = 2 x 2

P	**Q**	**(P ∧ Q)**	**~(P ∧ Q)**

Starting with the rightmost sentence letter – here, “Q” – we alternate single “1s” and “0s” under “Q” for the required number of times.

4 = 2 x 2

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

And then, for each sentence letter to the left, we double how many “1s” or “0s” we add at a time. So with “P” we put *two* “1s” in a row, then two “0s,” for the required number of valuations – 4.

4 = 2 x 2

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

We move next to “(P ∧ Q)”. Like all conjunctions, “(P ∧ Q)” follows the semantic rule for conjunctions.

Conjunction Rule:

●	▲	(● ∧ ▲)
1	1	1
1	0	0
0	1	0
0	0	0

When both parts are true – Valuation 1 – the whole conjunction is **true**.

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

When both parts aren’t true together – Valuations 2, 3, and 4 – the whole conjunction is **false**.

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

We next build the truth table for "~(P ∧ Q)". Since "~(P ∧ Q)" is a negation, it follows the Negation Rule.

Negation Rule:

●	~●
1	0
0	1

When "(P ∧ Q)" is true – Valuation 1 – its negation, "~(P ∧ Q)," will be false.

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

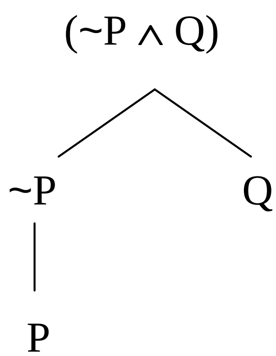
When "(P ∧ Q)" is false – Valuations 2, 3, and 4 – its negation "~(P ∧ Q)" is true.

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

That completes the truth table for the sentence "~(P ∧ Q)".

Sentence (2): $(\sim P \wedge Q)$

Discussion: Like the previous sentence, this sentence has a tilde and a wedge. But here the leftmost symbol is a parenthesis – which is brought by a wedge, but not by a tilde. So the wedge must be the main connective, making the sentence a conjunction. As the construction tree makes clear, the left part of the conjunction is "~P," and the right part is "Q"



The truth table for this sentence mirrors the construction.

P	**Q**	**~P**	**(~P ∧ Q)**

One again, by putting a "2" above each sentence letter, and multiplying the "2s," we calculate the number of valuations needed to cover all the possibilities: 4.

4 = 2 x 2

P	**Q**	**~P**	**(~P ∧ Q)**

And we build the valuations under "P" and "Q" exactly as in the last problem.

4 = 2 x 2

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

"~P" is a negation, following the Negation Rule.

Negation Rule:

●	~●
1	0
0	1

When "P" is true – Valuations 1 and 2 – "~P" is false.

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

When "P" is false – Valuations 3 and 4 – "~P" is true.

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

Next "(~P ∧ Q)" is built up from its two parts, "~P" and "Q". Since "(~P ∧ Q)" is a conjunction, it follows the Conjunction Rule: a conjunction is true only when both its parts are true. "~P" and "Q" are both true only in Valuation 3.

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

"(~P ∧ Q)" is false in the other valuations.

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

Note, finally, that the sentences "~(P ∧ Q)" and "(~P ∧ Q)"do **not** have the same truth table. So they are **not *logically equivalent***.

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

Since logical equivalence acts as a measure of when sentences have the same (logical) meaning, we conclude that these sentences ***do not*** (logically) ***mean the same thing***.

If we consider English counterparts to these two sentences, this verdict makes sense.

Letting “P” stand for “*We’re having ice cream*,” and “Q” stand for “*We’re having cake*,” the first sentence, “(~P ∧ Q),” would mean “***We’re not having ice cream, but we’re having cake***”.

P: We’re having ice cream
Q: We’re having cake

(~P ∧ Q): “We’re not having ice cream, but we’re having cake.”

Using the same translation table, an English counterpart to the second formal sentence can be approached in steps. First, the English sentence “*We’ll have both ice cream and cake*” is translated as “(P ∧ Q)”. So its negation, “***It is not the case that we’re having both ice cream and cake***” – more naturally, “***We’re not having* both *ice cream* and *cake***” – would have as its logical form our second formal sentence, “~(P ∧ Q)”.

P: We’re having ice cream
Q: We’re having cake

(~P ∧ Q): “We’re not having ice cream, but we’re having cake.”
~(P ∧ Q): “We’re not having *both* ice cream *and* cake.”

As English speakers we agree that these two sentences **don’t** mean the same thing. So our truth tables yield the right results for these two sentences.