Truth Table

p	**q**	**p∨q**	**¬(p∧q)**	**(p∨q)∧¬(p∧q)**
T	T	T	F	F
T	F	T	T	T
F	T	T	T	T
F	F	F	T	F

We want to find the inverse of this **(p∨q)∧¬(p∧q)** which would be finding the inverse of the “exclusive or”.

I believe **¬(p∨q)∨(p∧q)** works but I’m wondering if it’s the simplest possible case.

Is there a way to go from conjunctions (∧) to disjunctions (∨) using nots (**¬**)?

(p∨q)∧¬(p∧q)	**¬(p∨q)∨(p∧q)**

¬(p∨q)	**p∧q**	**¬(p∨q)∨(p∧q)**
F	T	T
F	F	F
F	F	F
T	F	T

What happens when we bring recursion into this? i.e. we plug entire statements in as p or q recursively. What happens to the truth tables?

¬(p∧q)	p∨q	¬(p∨q)	p∧q
F	T	F	T
T	T	F	F
T	T	F	F
T	F	T	F

Original	¬ Distributed	∨ -> ∧	Both
¬(p∨q)∨(p∧q)	(p∨q)∨ ¬(p∧q)	¬(p∨q)∧(p∧q)	(p∨q)∧ ¬(p∧q)
T	T	F	F
F	T	F	T
F	T	F	T
T	T	F	F

If you distribute, as done above, do you get all true for all statements? Or is dependent on the original truth table? If you go from conjunction to disjunction do they all go to false or once again does this depend on the statement?

For imply statements, when we have p→q, and p is false and q is true, it seems like p could imply q but it does not necessarily. So why do we say that the implication is true in that case?

Is it true that if you distribute and alternate between conjunction and disjunction all truth table values will alternate?

Why is this true?

$p \rightarrow q$ is logically equivalent to $\neg p \vee q$