Tautologies, Contingencies and Contradictions

A compound proposition that is true, no matter the truth values of its atomic propositions, is called a tautology. For example, since P must be either true or false $P \vee (\neg P)$ is always true.

Tautologies, Contingencies and Contradictions

A compound proposition that is true, no matter the truth values of its atomic propositions, is called a tautology. For example, since P must be either true or false $P \vee (\neg P)$ is always true.

A compound proposition that is always false, no matter the truth values of its atomic propositions, is called a contradiction. For example, $P \wedge (\neg P)$.

Tautologies, Contingencies and Contradictions

A compound proposition that is true, no matter the truth values of its atomic propositions, is called a tautology. For example, since P must be either true or false $P \vee (\neg P)$ is always true.

A compound proposition that is always false, no matter the truth values of its atomic propositions, is called a contradiction. For example, $P \wedge (\neg P)$.

A compound proposition that is sometimes true and sometimes false is called a contingency.

An Example . . . of One of Those

Consider the example

$$(\neg(A \wedge B)) \Leftrightarrow ((\neg A) \vee (\neg B))$$

.

An Example . . . of One of Those

Consider the example

$$(\neg(A \wedge B)) \Leftrightarrow ((\neg A) \vee (\neg B))$$

this is a tautology

.

A	B	$A \wedge B$	$\overset{C}{\neg(A \wedge B)}$	$\neg A$	$\neg B$	$\overset{D}{(\neg A) \vee (\neg B)}$	$(C) \Leftrightarrow (D)$
0	0	0	1	1	1	1	1
0	1	0	1	1	0	1	1
1	0	0	1	0	1	1	1
1	1	1	0	0	0	0	1

Logical Equivalence

When we considered $P \Rightarrow Q$ and its contrapositive $(\neg Q) \Rightarrow (\neg P)$, we came to the conclusion that they mean the same thing.

$$(P \Rightarrow Q) \Leftrightarrow ((\neg Q) \Rightarrow (\neg P))$$

is a tautology

Logical Equivalence

When we considered $P \Rightarrow Q$ and its contrapositive $(\neg Q) \Rightarrow (\neg P)$, we came to the conclusion that they mean the same thing. We can be precise about what we mean by that by saying that two propositions are logically equivalent if they have the same truth values for any choice of truth values of their combined atomic propositions.

Logical Equivalence

When we considered $P \Rightarrow Q$ and its contrapositive $(\neg Q) \Rightarrow (\neg P)$, we came to the conclusion that they mean the same thing. We can be precise about what we mean by that by saying that two propositions are logically equivalent if they have the same truth values for any choice of truth values of their combined atomic propositions.

Notice that we just showed that $\neg(P \wedge Q)$ is logically equivalent to $(\neg P) \vee (\neg Q)$.

Logical Equivalence

When we considered $P \Rightarrow Q$ and its contrapositive $(\neg Q) \Rightarrow (\neg P)$, we came to the conclusion that they mean the same thing. We can be precise about what we mean by that by saying that two propositions are logically equivalent if they have the same truth values for any choice of truth values of their combined atomic propositions.

Notice that we just showed that $\neg(P \wedge Q)$ is logically equivalent to $(\neg P) \vee (\neg Q)$. We can see from that that $\Leftrightarrow$ is essentially the symbol for logical equivalence.

Logical Equivalence

When we considered $P \Rightarrow Q$ and its contrapositive $(\neg Q) \Rightarrow (\neg P)$, we came to the conclusion that they mean the same thing. We can be precise about what we mean by that by saying that two propositions are logically equivalent if they have the same truth values for any choice of truth values of their combined atomic propositions.

Notice that we just showed that $\neg(P \wedge Q)$ is logically equivalent to $(\neg P) \vee (\neg Q)$. We can see from that that $\Leftrightarrow$ is essentially the symbol for logical equivalence. Although, sometimes, if we want to talk about the logical equivalence being "*about* the logic instead of *in* the logic" we'll use $\equiv$.

An Example from Programming

As another example consider the following code:

...

IF $(\overset{A}{(x > 0)} || (\overset{\neg A}{(x <= 0)} \&\& \overset{B}{(y > 100)}))$ THEN

...

...

A	B	$A \vee ((\neg A) \wedge B)$
0	0	0
0	1	1
1	0	1
1	1	1

An Example from Programming

As another example consider the following code:

$\ldots$

IF $((x > 0)||((x <= 0)\&\&(y > 100)))$ THEN

$\ldots$

$\ldots$

If we name $x > 0$ "A" and name $y > 100$ "B" than the following truth table shows that $A \vee (\neg A \wedge B)$ is logically equivalent to just $A \vee B$.

A	B	$A \vee (\neg A \wedge B)$	$A \vee B$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

Order of Operations

Just as in arithmetic, we can reduce our use of braces by deciding on an order of operations for the common logical connectives. The one in force is:

Order of Operations

Just as in arithmetic, we can reduce our use of braces by deciding on an order of operations for the common logical connectives. The one in force is:

1. just as in arithmetic, first resolve anything inside braces,

Order of Operations

Just as in arithmetic, we can reduce our use of braces by deciding on an order of operations for the common logical connectives. The one in force is:

1. just as in arithmetic, first resolve anything inside braces,
2. then apply $\neg$,

Order of Operations

Just as in arithmetic, we can reduce our use of braces by deciding on an order of operations for the common logical connectives. The one in force is:

1. just as in arithmetic, first resolve anything inside braces,
2. then apply $\neg$,
3. after which handle $\wedge$,

Order of Operations

Just as in arithmetic, we can reduce our use of braces by deciding on an order of operations for the common logical connectives. The one in force is:

1. just as in arithmetic, first resolve anything inside braces,
2. then apply $\neg$,
3. after which handle $\wedge$,
4. then any $\vee$'s (and $\oplus$'s),

Order of Operations

Just as in arithmetic, we can reduce our use of braces by deciding on an order of operations for the common logical connectives. The one in force is:

1. just as in arithmetic, first resolve anything inside braces,
2. then apply $\neg$,
3. after which handle $\wedge$,
4. then any $\vee$'s (and $\oplus$'s),
5. next comes $\Rightarrow$,

Order of Operations

Just as in arithmetic, we can reduce our use of braces by deciding on an order of operations for the common logical connectives. The one in force is:

1. just as in arithmetic, first resolve anything inside braces,
2. then apply $\neg$,
3. after which handle $\wedge$,
4. then any $\vee$'s (and $\oplus$'s),
5. next comes $\Rightarrow$,
6. and finally $\Leftrightarrow$.

Order of Operations

Just as in arithmetic, we can reduce our use of braces by deciding on an order of operations for the common logical connectives. The one in force is:

1. just as in arithmetic, first resolve anything inside braces,
2. then apply $\neg$,
3. after which handle $\wedge$,
4. then any $\vee$'s (and $\oplus$'s),
5. next comes $\Rightarrow$,
6. and finally $\Leftrightarrow$.

By the way, $\wedge$ is also called a conjunction,

Order of Operations

Just as in arithmetic, we can reduce our use of braces by deciding on an order of operations for the common logical connectives. The one in force is:

1. just as in arithmetic, first resolve anything inside braces,
2. then apply $\neg$,
3. after which handle $\wedge$,
4. then any $\vee$'s (and $\oplus$'s),
5. next comes $\Rightarrow$,
6. and finally $\Leftrightarrow$.

By the way, $\wedge$ is also called a conjunction, and $\vee$ is called a disjunction.

Arguments

Another use of truth tables is to evaluate arguments.

Arguments

Another use of truth tables is to evaluate arguments. In logic, an argument is a series of statements, called the premises, followed by one more proposition, called the conclusion.

Arguments

Another use of truth tables is to evaluate arguments. In logic, an argument is a series of statements, called the premises, followed by one more proposition, called the conclusion. An argument is called valid if the conclusion is true whenever the premises are all true.

Arguments

Another use of truth tables is to evaluate arguments. In logic, an argument is a series of statements, called the premises, followed by one more proposition, called the conclusion. An argument is called valid if the conclusion is true whenever the premises are all true. If any of the premises are not true than the truth value of the conclusion in that row is unimportant.

Arguments

Another use of truth tables is to evaluate arguments. In logic, an argument is a series of statements, called the premises, followed by one more proposition, called the conclusion. An argument is called valid if the conclusion is true whenever the premises are all true. If any of the premises are not true than the truth value of the conclusion in that row is unimportant.

The argument that will be familiar to most of you is the famous syllogism:

$$\begin{array}{l} \text{Socrates is a man} \\ \text{All men are mortal} \\ \hline \therefore \text{ Socrates is mortal} \end{array}$$

An Example of an Argument

Let us determine if the following argument is valid

$$\begin{array}{l} P \Rightarrow (Q \vee (\neg R)) \\ Q \Rightarrow (P \wedge R) \\ \hline \therefore P \Rightarrow R \end{array}$$

An Example of an Argument

Let us determine if the following argument is valid

$$\begin{array}{l} P \Rightarrow (Q \vee (\neg R)) \\ Q \Rightarrow (P \wedge R) \\ \hline \therefore P \Rightarrow R \end{array}$$

We need to make sure that the conclusion $(P \Rightarrow R)$ is true whenever all the premises are true.

$$\Big(\big(P \Rightarrow (Q \vee \neg R)\big) \wedge \big(Q \Rightarrow (P \wedge R)\big)\Big) \Rightarrow (P \Rightarrow R)$$

The Truth Table Work

premise 1 premise 2

This row tells us the argument is invalid

P	Q	R	$Q \vee \neg R$	$P \wedge R$	$P \Rightarrow (Q \vee \neg R)$	$Q \Rightarrow (P \wedge R)$	$P \Rightarrow R$
0	0	0	1	0	1	1	1
0	0	1	0	0	1	1	1
0	1	0	1	0	1	0	X
0	1	1	1	0	1	0	X
1	0	0	1	0	1	1	0
1	0	1	0	1	0	1	X
1	1	0	1	0	1	0	X
1	1	1	1	1	1	1	1

This row can't make the argument invalid

Logical Rules

First there are two standard basic arguments:

modus ponens

("mode that affirms by affirming")

$$\begin{array}{c} P \Rightarrow Q \\ P \\ \hline \therefore Q \end{array}$$

and

modus tollens

("mode that denies by denying")

$$\begin{array}{c} P \Rightarrow Q \\ \neg Q \\ \hline \therefore \neg P \end{array}$$