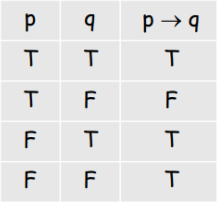
Lecture 14. Logical Equivalence

The converse, contrapositive, and inverse

A few statements related to $p \to q$:

- The converse of $p \to q$ is $q \to p$.
- The contrapositive of $p \to q$ is $\neg q \to \neg p$.
- The inverse of $p \to q$ is $\neg p \to \neg q$.

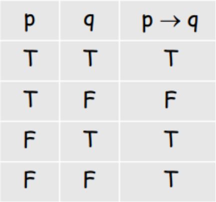


p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

The converse, contrapositive, and inverse

A few statements related to $p \to q$:

- The converse of $p \to q$ is $q \to p$.
- The contrapositive of $p \to q$ is $\neg q \to \neg p$.
- The inverse of $p \to q$ is $\neg p \to \neg q$.



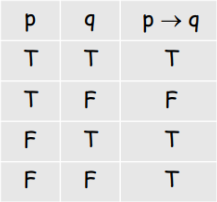
p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

Example 14.1. Given the statement, “If graph G is bipartite, it cannot contain an odd length cycle. .”, describe its converse, contrapositive, and inverse.

The converse, contrapositive, and inverse

A few statements related to $p \rightarrow q$:

- The converse of $p \rightarrow q$ is $q \rightarrow p$.
- The contrapositive of $p \rightarrow q$ is $\neg q \rightarrow \neg p$.
- The inverse of $p \rightarrow q$ is $\neg p \rightarrow \neg q$.



p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

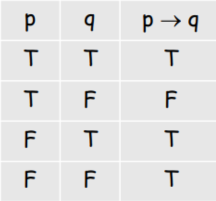
Example 14.1. Given the statement, “If graph G is bipartite, it cannot contain an odd length cycle. .”, describe its converse, contrapositive, and inverse.

- Converse: If graph G doesn’t contain an odd length cycle, it can be bipartite.

The converse, contrapositive, and inverse

A few statements related to $p \rightarrow q$:

- The converse of $p \rightarrow q$ is $q \rightarrow p$.
- The contrapositive of $p \rightarrow q$ is $\neg q \rightarrow \neg p$.
- The inverse of $p \rightarrow q$ is $\neg p \rightarrow \neg q$.



p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

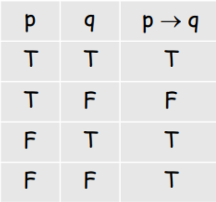
Example 14.1. Given the statement, "If graph G is bipartite, it cannot contain an odd length cycle. .", describe its converse, contrapositive, and inverse.

- Converse: If graph G doesn't contain an odd length cycle, it can be bipartite.
- Contrapositive: If graph G contain an odd length cycle, it is not bipartite.

The converse, contrapositive, and inverse

A few statements related to $p \to q$:

- The converse of $p \to q$ is $q \to p$.
- The contrapositive of $p \to q$ is $\neg q \to \neg p$.
- The inverse of $p \to q$ is $\neg p \to \neg q$.



p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

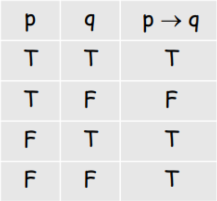
Example 14.1. Given the statement, "If graph G is bipartite, it cannot contain an odd length cycle. .", describe its converse, contrapositive, and inverse.

- Converse: If graph G doesn't contain an odd length cycle, it can be bipartite.
- Contrapositive: If graph G contain an odd length cycle, it is not bipartite.
- Inverse: If graph G is not bipartite, it can contain an odd length cycle.

The converse, contrapositive, and inverse

A few statements related to $p \to q$:

- The converse of $p \to q$ is $q \to p$.
- The contrapositive of $p \to q$ is $\neg q \to \neg p$.
- The inverse of $p \to q$ is $\neg p \to \neg q$.



p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

Example 14.1. Given the statement, “If graph G is bipartite, it cannot contain an odd length cycle. .”, describe its converse, contrapositive, and inverse.

- Converse: If graph G doesn’t contain an odd length cycle, it can be bipartite.
- Contrapositive: If graph G contain an odd length cycle, it is not bipartite.
- Inverse: If graph G is not bipartite, it can contain an odd length cycle.

One of the three is equivalent to the original conditional statement, WHICH ONE? How do you know?

Logical Equivalence

Two statements p and q are logically equivalent, denoted by $p \equiv q$, if and only if they have the same truth values for all possible combinations of truth values for the propositional variables.

Logical Equivalence

Two statements p and q are logically equivalent, denoted by $p \equiv q$, if and only if they have the same truth values for all possible combinations of truth values for the propositional variables.

p	q	$p \rightarrow q$	$q \rightarrow p$

$\neg p$	$\neg q$	$\neg q \rightarrow \neg p$	$\neg p \rightarrow \neg q$

Logical Equivalence

Two statements p and q are logically equivalent, denoted by $p \equiv q$, if and only if they have the same truth values for all possible combinations of truth values for the propositional variables.

p	q	$p \to q$	$q \to p$

$\neg p$	$\neg q$	$\neg q \to \neg p$	$\neg p \to \neg q$

- The contrapositive is equivalent to the original conditional statement:

$$p \to q \Leftrightarrow \neg q \to \neg p$$

- The converse and inverse are equivalent to each other.

$$q \to p \Leftrightarrow \neg p \to \neg q$$

Example 14.2. Consider the following theorem from secondary school geometry.

> *If a quadrilateral has a pair of parallel sides, then it has a pair of supplementary angles.* [1]

This theorem is of the form $p \rightarrow q$. Determine p and q and write the theorem in its contropositive form, $\neg q \rightarrow \neg p$, which is logically equivalent to $p \rightarrow q$

[1] Recall that two angles are supplementary if their angle measures sum to 180°

Example 14.2. Consider the following theorem from secondary school geometry.

> *If a quadrilateral has a pair of parallel sides, then it has a pair of supplementary angles.* [1]

This theorem is of the form $p \to q$. Determine p and q and write the theorem in its contropositive form, $\neg q \to \neg p$, which is logically equivalent to $p \to q$

Practice 14.3. Determine if the following compound statements are logically equivalent.

(a) $p \to q$ and $\neg p \vee q$

(b) $\neg(p \to q)$ and $\neg p \to \neg q$

[1] Recall that two angles are supplementary if their angle measures sum to 180°

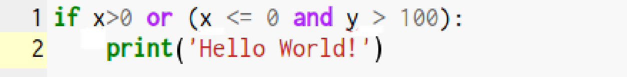
Propositional logic in computer programs

Statements and logical connectives arise all the time in computer programs.

Propositional logic in computer programs

Statements and logical connectives arise all the time in computer programs.

Example 14.4. Consider the following snippet in Python code



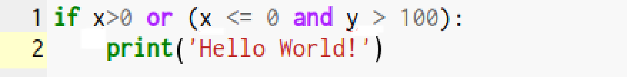
```
if x>0 or (x <= 0 and y > 100):
    print('Hello World!')
```

- The **print** statement is carried out when the *proposition* following the word **if** is *True*.
- On closer inspection, this big expression is built from two simpler propositions.

Propositional logic in computer programs

Statements and logical connectives arise all the time in computer programs.

Example 14.4. Consider the following snippet in Python code



```python
if x>0 or (x <= 0 and y > 100):
    print('Hello World!')
```

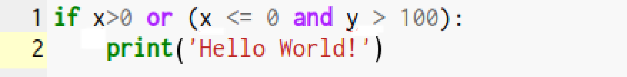
- The **print** statement is carried out when the *proposition* following the word **if** is *True*.
- On closer inspection, this big expression is built from two simpler propositions.

  A: $x > 0$
  B: $y > 100$

Propositional logic in computer programs

Statements and logical connectives arise all the time in computer programs.

Example 14.4. Consider the following snippet in Python code



```
if x>0 or (x <= 0 and y > 100):
    print('Hello World!')
```

- The **print** statement is carried out when the *proposition* following the word **if** is *True*.
- On closer inspection, this big expression is built from two simpler propositions.

  A: $x > 0$

  B: $y > 100$
- Then we can rewrite the 'if' condition as $A \vee (\neg A \wedge B)$.
- A truth table reveals that this complicated expression is logically equivalent to (what?).

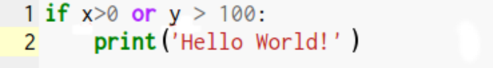
- (Continue from the previous example) ... $A \vee (\neg A \wedge B)$. is logically equivalent to (what?).

A	B	$\neg A$	$\neg A \wedge B$	$A \vee (\neg A \wedge B)$	(What?)
T	T				
T	F				
F	T				
F	F				

- (Continue from the previous example) ... $A \lor (\neg A \land B)$. is logically equivalent to (what?).

A	B	$\neg A$	$\neg A \land B$	$A \lor (\neg A \land B)$	(What?)
T	T				
T	F				
F	T				
F	F				

- This means that we can simplify the code snippet without changing the program's behavior:

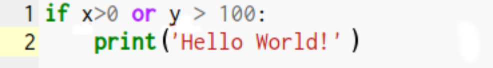


```
if x>0 or y > 100:
    print('Hello World!')
```

- (Continue from the previous example) ... $A \vee (\neg A \wedge B)$. is logically equivalent to (what?).

A	B	$\neg A$	$\neg A \wedge B$	$A \vee (\neg A \wedge B)$	(What?)
T	T				
T	F				
F	T				
F	F				

- This means that we can simplify the code snippet without changing the program's behavior:



```
if x>0 or y > 100:
    print('Hello World!')
```

Remarks:

- Rewriting a logical expression involving many variables in the simplest form is both difficult and important.
- Simplifying expressions in software can increase the speed of your program.

MSC Activity 14.5. Consider the proposition “If AI takes over the World or outsmarts people, AI will get the ability to feel emotion.” Write “E” for each proposition that is logically equivalent to the given proposition, “C” for each proposition that is logically equivalent to the converse of the given proposition and “N” if neither.

(a) Unless AI gains the ability to feel an emotion, AI will not dominate the world and cannot surpass humans.

(b) AI does not take over the World and outsmart people, or AI will get the ability to feel emotion.

(c) AI takes over the World or outsmarts people, and AI will not get the ability to feel an emotion.

(d) AI will not get the ability to feel an emotion if AI does not take over the World and outsmart people.

(e) AI does not take over the World and outsmart people, and AI will not get the ability to feel an emotion.