1.1: Compound Statements

We can make a new statement from other statements; we call these **compound propositions** or **compound statements**.

**Example (PageIndex{1}):**

1. It is not the case that all birds can fly. (This is the negation of the statement all birds can fly).
2. \((1+1=2)\) and "All birds can fly". (Here the connector "and" was used to create a new statement).

Note the following four basic ways to start with one or more propositions and use them to make a more elaborate compound statement. If \((p)\) and \((q)\) are statements then here are four compound statements made from them:

1. \(\neg p\), Not \(p\) (i.e. the negation of \(p\)),
2. \((p \land q, \ p \ \text{and} \ q)\),
3. \((p \lor q, \ p \ \text{or} \ q)\) and
4. \((p \rightarrow q, \ \text{If} \ p \ \text{then} \ q)\)

**Example (PageIndex{2}):**

If \((p = \text{"You eat your supper tonight"})\) and \((q = \text{"You get desert"})\). Then

1. Not \((p)\) is "You don't eat your supper tonight".
2. \((p, \ \text{"You eat your supper tonight and you get desert"})\).
3. \((p \lor q, \ \text{return} \ \text{or} \ q)\) is "You eat your supper tonight or you get desert".
4. \((\text{If} \ p \ \text{then} \ q)\) is "If you eat your supper tonight then you get dessert."
In English, we know these four propositions don't say the same thing. In logic, this is also the case, but we can make that clear by displaying the truth value possibilities. It is common to use a table to capture the possibilities for truth values of compound statements. We call such a table a truth table. Below are the possibilities: the first is the least profound. It says that a statement \( p \) is either true or false.

\[
\begin{array}{c|c}
\neg(p) & T F \\
\hline
T & F \\
F & T \\
\end{array}
\]

**Negation**

Truth tables are more useful in describing the possible truth values for various compound propositions. Consider the following truth table:

\[
\begin{array}{c|c}
(p) & \neg(p) \\
\hline
T & F \\
F & T \\
\end{array}
\]

The table above describes the truth value possibilities for the statements \( (p) \) and \( \neg(p) \), or "not \( p \)". As you can see, if \( (p) \) is true then \( \neg(p) \) is false and if \( (p) \) false, the negation (i.e. not \( p \)) is true. \( \neg \) is the mathematical notation used to mean "not."

**Example \( \PageIndex{3} \):**

Consider the statement \( (p) \): \( 1 + 1 = 3 \).

Statement \( (p) \) can either be true or false, not both.

\( \neg(p) \) is "not \( (p) \)," or the negation of statement \( (p) \).

\( \neg(p) \) is \( (1 + 1 \neq 3) \).

You can see that the negation of a proposition affects only the proposition itself, not any other assumptions.

**Conjunction**

Conjunction statements use two or more propositions. If two or more simple propositions are involved the truth table gets bigger. Below is the truth table for "and," otherwise known as a conjunction. When is an and statement true? As the truth table
indicates, only when both of the component propositions are true is the compound conjunction statement true:

\[
\begin{array}{ccc}
(p) & (q) & (p \land q) \\
(T) & (T) & (T) \\
(T) & (F) & (F) \\
(F) & (T) & (F) \\
(F) & (F) & (F) \\
\end{array}
\]

Example (PageIndex{4}): 

Consider statements \(p:=1 + 1 = 2\) and \(q:=2 < 5\).

Note that, \((p \land q)\) is true only if both \(p\) and \(q\) are both true.

Since statements \(p\) and \(q\) are both true, \((p \land q)\) is true.

**Disjunction**

Disjunction statements are compound statements made up of two or more statements and are true when one of the component propositions is true. They are called "Or Statements." In English, "or" is used in two ways:

1. If a person is looking for a house with 4 bedrooms or a short commute, a real estate agent might present houses with either 4 bedrooms or a short commute or both 4 bedrooms and a short commute. This is called an **inclusive or**.
2. If a person is asked whether they would like a Coke or a Pepsi, they are expected to choose between the two options. This is an **exclusive or**: "both" is not an acceptable case.

In logic, we use **inclusive or** statements

\[
\begin{array}{ccc}
(p) & (q) & (p \lor q) \\
(T) & (T) & (T) \\
(T) & (F) & (T) \\
(F) & (T) & (T) \\
(F) & (F) & (F) \\
\end{array}
\]

The \((p \lor q)\) proposition is only false if both component propositions \((p)\) and \((q)\) are false.
Example (PageIndex{5}):

Consider the statement \( (2 \leq -3) \)

The statement reads "2 is less than or equal to -3", or "\((2 < -3 \vee 2 = -3)\)" and can be broken into two component propositions:

1. Proposition \((p)\): \( (2 < -3) \) (False)
2. Proposition \((q)\): \( (2 = -3) \) (False)

Because propositions \((p)\) and \((q)\) are both false, the statement is false.

Example (PageIndex{6}):

Consider the statement \( (2 \leq 5) \)

The statement's two component propositions are:

1. Proposition \((p)\): \( (2 < 5) \) (True)
2. Proposition \((q)\): \( (2 = 5) \) (False)

Since proposition \((p)\) is true, the statement is true.

### Conditional Statements

Consider the "if p then q" proposition. This is a conditional statement. Read the statements below. If these statements are made, in which instance is one lying (i.e. when is the overall statement false)?

Suppose, at suppertime, your mother makes the statement “If you eat your broccoli then you’ll get dessert.” Under what conditions could you say your mother is lying?

1. If you eat your broccoli but don't get dessert, she lied!
2. If you eat your broccoli and get dessert, she told the truth.
3. If you don’t eat your broccoli and you don’t get dessert she told you the truth.
4. If you don’t eat your broccoli but you do get dessert we still think she told the truth. After all, she only outlined one condition that was supposed to get you dessert, she didn’t say that was the only way you could earn dessert. Maybe you had cauliflower instead.

Note that the order in which the cases are presented in the truth table is irrelevant. The cases themselves are important information, not their order relative to each other.

\[
\begin{array}{ccc}
\neg(p) & \neg(q) & (p \to q) \\
T & F & F \\
F & T & F \\
F & F & T \\
\end{array}
\]
\( p \to q \)

<table>
<thead>
<tr>
<th>( p )</th>
<th>( q )</th>
<th>( p \to q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
</tr>
<tr>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
</tr>
<tr>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
</tr>
<tr>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
</tr>
<tr>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
</tr>
</tbody>
</table>

It is important to notice that, if the first proposition is false, the conditional statement is true by default. A conditional statement is defined as being true unless a true hypothesis leads to a false conclusion.

**Example (PageIndex{7}):**

Consider the statement "If a closed figure has four sides, then it is a square." This is a false statement - why?

We can prove it using a **counter-example**: we draw a four-sided figure that is not a square. So there!

**Example (PageIndex{8}):**

Consider the statement "If \(2 = 3\), then \(5 = 2\)"

Since \(2 \neq 3\), it does not matter if \(5 = 2\) is true or not, the conditional statement as a whole is true.

---

**Converse of a conditional statement**

Let \( P \) be a statement if \( p \) then \( q \). Then the converse of \( P \) is if \( q \) then \( p \).

**Example (PageIndex{9}):**

Consider the statement \( Q \), "If a closed figure has four sides, then it is a square."

Then the converse of \( Q \) is "If it is a square then it is a closed figure with four sides".

---

**Contrapositive of a Conditional Statement**

Let \( P \) be a statement if \( p \) then \( q \). Then the contrapositive of \( P \) is if \( \neg q \) then \( \neg p \).

**Example (PageIndex{10}):**

Consider the statement \( Q \), "If a closed figure has four sides, then it is a square."

Then the converse of \( Q \) is "If it is not a square then it is not a closed figure with four sides".

---

UC Davis ChemWiki is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License.
**Bi-Conditional Statements**

**Bi-conditional statements** are conditional statements which depend on both component propositions. They read "p if and only if q" and are denoted \((p \leftrightarrow q)\) or "p iff q", which is logically equivalent to \(((p \to q) \wedge (q \to p))\). These compound statements are true if both component propositions are true or both are false:

<table>
<thead>
<tr>
<th>(p)</th>
<th>(q)</th>
<th>(p \leftrightarrow q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Example \(\PageIndex{11}\):

Consider the statement: "Two lines are perpendicular if and only if they intersect to form a right angle."

The component propositions are:

1. \(p\): Two lines are perpendicular
2. \(q\): The lines intersect to form a right angle

Logically, we can see that if two lines are perpendicular, then they must intersect to form a right angle. Also, we can see that if two lines form a right angle, then they are perpendicular.

If two lines are not perpendicular, then they cannot form a right angle. Conversely, if two lines do not form a right angle, they cannot be perpendicular. This is why, if both propositions in a biconditional statement are false, the statement itself is true!

**Logically Equivalent Statements**

Once we know the basic statement types and their truth tables, we can derive the truth tables of more elaborate compound statements. Below is the truth table for the proposition, not \(p\) or \((p \wedge q)\). First, we calculate the truth values for not \(p\), then \(p\) and \(q\) and finally, we use these two columns of truth values to figure out the truth values for not \(p\) or \((p \wedge q)\).

<table>
<thead>
<tr>
<th>(p)</th>
<th>(q)</th>
<th>(\neg p)</th>
<th>(p \wedge q)</th>
<th>(\neg p \vee (p \wedge q))</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
So the proposition "not p or (p and q)" is only false if p is true and q is false. Does this seem familiar?

"If p then q" is only false if p is true and q is false as well.

This has some significance in logic because if two propositions have the same truth table they are in a logical sense equal to each other – and we say that they are **logically equivalent**. So: \( \neg p \vee (p \wedge q) \equiv p \to q \), or "Not p or (p and q) is equivalent to if p then q."

**Example \( \PageIndex{12} \):**

Prove or disprove: for any mathematical statements \( p, q \) and \( r \), \( p \to (q \vee r) \) is logically equivalent to \( \neg r \to (p \to q). \)

Hence, \( p \to (q \vee r) \) is logically equivalent to \( \neg r \to (p \to q) \).
Tautologies and Contradictions

There are two cases in which compound statements can be made that result in either always true or always false. These are called **tautologies** and **contradictions**, respectively. Let's consider a tautology first, and then a contradiction:

**Example (PageIndex{13}):**

Consider the statement "\((2 = 3) \vee (2 \neq 3)\):"

There are two component propositions:

1. \(p\): \(2 = 3\)
2. \(\neg p\): \(2 \neq 3\)

Clearly, this statement is a tautology.

Let's make a truth table for general case \(p \vee (\neg p)\):

<table>
<thead>
<tr>
<th>(p)</th>
<th>(\neg p)</th>
<th>(p \vee (\neg p))</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

As you can see, no matter what we do, this statement is always true. It is a **tautology**. Careful! This is not to say that this statement makes logical sense in English, but rather that, using logical mathematics, this statement is always true.

**Example (PageIndex{14}):**

Consider the statement "2 is even \(\wedge\) 2 is odd"\

There are two component propositions:

1. \(p\): 2 is even
2. \(\neg p\): 2 is odd

Clearly this statement is a contradiction.

Let's make a truth table for general case \(p \wedge (\neg p)\):

<table>
<thead>
<tr>
<th>(p)</th>
<th>(\neg p)</th>
<th>(p \wedge (\neg p))</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>
\( \neg p \)
\( \neg (\neg p) \)
\( p \wedge (\neg p) \)

\( T \)
\( F \)
\( F \)

As you can see again, no matter what we do, this statement will always be false. It is a contradiction. These make more sense in English: 2 cannot be both even and odd, after all! Still, what matters is what we decide using logical mathematics.

**Summary**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Notation</th>
<th>Summary of truth values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negation</td>
<td>( \neg p )</td>
<td>Opposite truth value of ( p )</td>
</tr>
<tr>
<td>Conjunction</td>
<td>( p \wedge q )</td>
<td>True only when both ( p ) and ( q ) are true</td>
</tr>
<tr>
<td>Disjunction</td>
<td>( p \vee q )</td>
<td>False only when both ( p ) and ( q ) are false</td>
</tr>
<tr>
<td>Conditional</td>
<td>( p \to q )</td>
<td>False only when ( p ) is true and ( q ) is false</td>
</tr>
<tr>
<td>Biconditional</td>
<td>( p \leftrightarrow q )</td>
<td>True only when both ( p ) and ( q ) are true or both are false</td>
</tr>
</tbody>
</table>

**New Notations & Definitions**

- Negation: \( \neg \) or "not"
- Conjunction: \( \wedge \) or "and"
- Disjunction: \( \vee \) or "or"
- Conditional: \( \to \) or "implies" or "if/then"
- Bi-Conditional: \( \leftrightarrow \) or "if and only if" or "iff"
- Counter-example: An example that disproves a mathematical proposition or statement.
- Logically Equivalent: \( \equiv \) Two propositions that have the same truth table result.
- Tautology: A statement that is always true, and a truth table yields only true results.
- Contradiction: A statement which is always false, and a truth table yields only false results.