

## Section 1.12: The Biconditional

### The Biconditional Connective

Consider the conditional “If you live in Victoria, then you live in BC.” Remembering that the conditional has nothing to say if the first proposition is false, then it is possible for you to not live in Victoria but to still live in BC (Nanaimo, Vancouver, etc.). It is also possible for you to not live in Victoria and also not live in BC (Calgary, AB or Toronto, ON).

Let’s now consider the conditional “If the temperature outside is below 0°C, then it is freezing outside.” If I were to use this sentence in everyday English, I probably mean “If the temperature outside is below 0°C, then it is freezing outside **AND** if the temperature outside is **not** below 0°C, then it’s **not** freezing outside.” So we could probably do with a new proposition that means “If  $p$ , then  $q$  **and** if  $\bar{p}$ , then  $\bar{q}$ .” This connective is called the **biconditional**,  $p \leftrightarrow q$ .

The truth table for the biconditional, then, is

$p$	$q$	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

So if  $p$  and  $q$  have the same value, then the biconditional is true and otherwise it’s false. (In a sense, then, it’s the negation of “exclusive or”,  $p \oplus q$ .) There are a number of ways to specify the biconditional in English:

- If and only if  $p$ , then  $q$ .
- $p$  if and only if  $q$ .
- If  $p$ , then  $q$ , and vice versa.
- If  $p$ , then  $q$ , and if  $\bar{p}$ , then  $\bar{q}$ .

We’ll mostly be using the first construction, using “if and only if”.

### *Example*

Draw the truth tables for  $p \leftrightarrow q$  and  $(p \rightarrow q) \wedge (q \rightarrow p)$ . Are they logically equivalent? Also, are  $p \leftrightarrow q$  and  $(p \rightarrow q) \wedge (\bar{p} \rightarrow \bar{q})$  logically equivalent?

Answer: Let’s draw a big truth table:

p	q	$p \leftrightarrow q$	$\bar{p}$	$\bar{q}$	$p \rightarrow q$	$q \rightarrow p$	$\bar{p} \rightarrow \bar{q}$	$(p \rightarrow q) \wedge (q \rightarrow p)$	$(p \rightarrow q) \wedge (\bar{p} \rightarrow \bar{q})$
0	0	1	1	1	1	1	1	1	1
0	1	0	1	0	1	0	0	0	0
1	0	0	0	1	0	1	1	0	0
1	1	1	0	0	1	1	1	1	1

So by looking at the relevant columns, we can see that all three logical expressions are equivalent. So (“if p, then q” AND “if q, then p”) is equivalent to the biconditional. (We could have also noted that since the converse ( $q \rightarrow p$ ) is logically equivalent to the inverse ( $\bar{p} \rightarrow \bar{q}$ ), then the last two columns should be identical.)

### **Example**

Consider the following conditional statements.

- “If two lines are perpendicular, then the angle between them is  $90^\circ$ .”
- “If a polygon is a right triangle, then it has three sides.”

Which of these sentences would still be true if it were written in the form of the biconditional?

Answer: (a) would still be true since if two lines are not perpendicular, then the angle between them is not  $90^\circ$ . (b) would not be true, since there are many triangles that aren’t right triangles.

### **Example**

The following biconditional statement is true: “If and only if Pat finishes her marking, she will not feel guilty.” Given that, answer the following questions.

- Pat feels guilty. Did she finish her marking?
- Pat does not feel guilty. Did she finish her marking?
- Pat finished her marking. Does she feel guilty?
- Pat did not finish her marking. Does she feel guilty?

Answer: Let  $p$  = “Pat finishes her marking” and  $q$  = “Pat does not feel guilty.”

- So  $q$  is false, so  $p$  must also be false (the biconditional requires that  $p$  and  $q$  have the same values). So Pat did not finish her marking and the answer is “No.”
- $q$  is true so  $p$  is true. Yes.

- c)  $p$  is true so  $q$  is true. And since  $q$  is “not guilty”, the answer is No.
- d)  $p$  is false so  $q$  is false. And feeling “not-not guilty” is just “guilty”, so Yes.

### Programming Applications

The if-then construction is very common in programming. In pseudocode, it usually takes the form

```
if  $x > 3$  then print "Hello World"
```

When you are debugging, it is tempting to think that this particular code fragment behaves more like the biconditional: if “Hello World” was output, was  $x > 3$ ? It’s tempting to think so, but what if “Hello World” was printed because of some other command? What then can we conclude about  $x$ ?

Let’s examine this in more detail. Recall that if  $p \rightarrow q$  is true and  $q$  is true, we cannot conclude anything about  $p$ . Now consider the following piece of pseudocode.

```
 $x = 4$   
 $y = 0$   
if  $x > 3$  then  $y = 5$   
print "y = ",  $y$ 
```

The output will be “ $y = 5$ ”.

But, won’t the following pseudocode have the same output?

```
 $x = 2$   
 $y = 5$   
if  $x > 3$  then  $y = 5$   
print "y = ",  $y$ 
```

It will. The conditional statement did not change the value of  $y$ , but the value was set to 5 initially, so the output is still be “ $y = 5$ ”. Once again, knowing that  $q$  is true does not allow us to conclude anything about  $p$  from the conditional  $p \rightarrow q$ . However, if the output was “ $y = 4$ ” or any other value not equal to 5, we can draw the conclusion that  $x$  was **not** greater than 3. If  $p \rightarrow q$  is true and  $q$  is false, we know with certainty that  $p$  is false also.

The if-then-else construction behaves in a similar fashion. Consider the following code fragment:

```
if  $x > 3$  then  $y = 5$   
          else  $z = 7$   
print "y = ",  $y$ , "z = ",  $z$ 
```

Only if the output tells you that  $y \neq 5$  or  $z \neq 7$  will you know with certainty something about  $x$ .

For special cases, the if-then-else construction can yield more information. Consider the following.

```
if x > 3 then y = 5
           else y = 7
print "y = ", y
```

Since this piece of pseudocode assigns different values (5 or 7) to the **same** variable  $y$ , finding out the resulting value of  $y$  will determine whether  $x$  was greater than 3. In this special case, the if-then behaves like the biconditional: if  $y = 5$  then you know that  $x > 3$ , and if  $y \neq 5$ , then  $x \leq 3$ .