

## BOOLEAN LOGIC

James T. Smith  
San Francisco State University

You often make compound sentences by connecting shorter ones. Sometimes you can determine the truth or falsehood of the longer sentence just from knowledge of the truth or falsehood of its components. In that case, the compound is called *Boolean*. For example, sentence *B* below is a Boolean compound, but *C* is not:

Component <i>P</i> :	I'm wearing my helmet.
Component <i>Q</i> :	Biking is dangerous.
$B = "P \text{ and } Q"$ :	I'm wearing my helmet <i>and</i> biking is dangerous.
$C = "P \text{ because } Q"$ :	I'm wearing my helmet <i>be-</i> <i>cause</i> biking is dangerous.

Sentence *B* is true just when both components are true. But, even when *P* and *Q* are both true, you still can't verify *C* without knowing how I was thinking when I put on my helmet. (I might have donned it for safety or just because my wife said to.) Words or phrases like *and* and *because* that we use to build compound sentences are called *connectives*. Those that result in Boolean compounds are called *Boolean*. Thus, *and* is Boolean but *because* is not. The systematic study of Boolean connectives is called *Boolean logic*. Some equivalent terms are also common: *propositional*, *sentential*, or *truth functional* logic.

Besides *and*, several more Boolean connectives are common. Here are some examples:

Component <i>P</i> :	I'm wearing my helmet.
Component <i>Q</i> :	Biking is dangerous.
$B = "not\ P"$ :	I'm <i>not</i> wearing my helmet.
$B = "neither\ P\ nor\ Q"$ :	I'm <i>not</i> wearing my helmet <i>nor</i> is biking dangerous.
$B = "P\ if\ and\ only\ if\ Q"$ :	I wear my helmet <i>if and only if</i> biking is dangerous.
$B = "P\ or\ Q"$ :	I'm wearing my helmet <i>or</i> biking is dangerous.

Evidently, you must sometimes vary the wording a little to avoid awkward diction. These Boolean connectives occur so often that they have symbolic abbreviations and the corresponding compound sentences have English names:

Connective	Sym- bol	Com- pound	Name
<i>and</i>	$\&$	$P \& Q$	<i>Conjunction</i> of $P, Q$
<i>not</i>	$\neg$	$\neg P$	<i>Negation</i> of $P$
<i>neither/nor</i>	$\downarrow$	$P \downarrow Q$	(None is common.)
<i>if and only if</i>	$\Leftrightarrow$	$P \Leftrightarrow Q$	<i>Equivalence</i> of $P, Q$
<i>or</i>	$\vee$	$P \vee Q$	<i>Disjunction</i> of $P, Q$

(Various other symbols are commonly used, too.)

Since there are so many Boolean connectives, you could use a device to help determine the truth or falsehood of a Boolean compound once you know that of its components. Pause for a moment: the awkward phrase *truth or falsehood* will recur frequently in this discussion unless an abbreviation is adopted, so the term *truth value* will be used from now on. *Truth tables* are commonly used to determine the truth value of a Boolean compound from that of its components. Here are the tables for most of the examples considered so far. T and F stand for the truth values.

$P$	$Q$	$P \& Q$	$P$	$\neg P$	$P$	$Q$	$P \downarrow Q$	$P$	$Q$	$P \Leftrightarrow Q$
T	T	T	T	F	T	T	F	T	T	T
T	F	F	F	T	T	F	F	T	F	F
F	T	F	F	T	F	T	F	F	T	F
F	F	F	F	T	F	F	T	F	F	T

Using truth tables you can see clearly that a conjunction is true just when *both* components are, negation *reverses* truth values, and that an equivalence is true just when both components have the *same* truth value.

There's a problem with the truth table for disjunction: which of these should it be?

$P$	$Q$	$P \vee Q$	$P$	$Q$	$P \neq Q$
T	T	T	T	T	F
T	F	T	T	F	T
F	T	T	F	T	T
F	F	F	F	F	F

The first alternative is normally selected: a disjunction " $P$  or  $Q$ " is false just when both components are. The second alternative is usually rendered differently in

English: “*either P or Q*”. It’s given a different symbol  $\neq$  and a different name, *exclusive* disjunction.

An equivalence “*P if and only if Q*” is often regarded as the conjunction of two *conditional* sentences “if *P* then *Q*” and “if *Q* then *P*”. Is *if... then...* a Boolean connective? That is, does everyone agree on the truth value of “if *P* then *Q*” —abbreviated  $P \Rightarrow Q$  —once they concur about *P* and *Q*? In fact, there’s general agreement on only one line in the corresponding truth table:

$P$	$Q$	$P \Rightarrow Q$
T	T	?
T	F	F
F	T	?
F	F	?

The other cases are problematic. Suppose

$P$ :	I’m wearing my helmet
$Q$ :	Biking is dangerous

are both true. Is the conditional

$P \Rightarrow Q$  : If I’m wearing my helmet then biking is dangerous

true? You probably hesitate to answer, and you’d probably hesitate, too, if *P* were false. In each of these cases in everyday life, to determine the truth value of  $P \Rightarrow Q$ , you seem to need more information about *P* and *Q* than just their truth values. Conditional sentences, as used in everyday life, are not Boolean compounds.

However, conditional sentences play an important role in mathematics, and there they usually don’t cause the same perplexity they do in everyday life. Therefore, in mathematical practice, conditional sentences are frequently used as though  $\Rightarrow$  were a Boolean connective with the following truth table:

$P$	$Q$	$P \Rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

Thus, mathematicians generously regard conditionals as true except when inarguably refuted by the case in the second line.

There's an important sense in mathematics where  $\Rightarrow$  is used in a non-Boolean manner. Often  $P \Rightarrow Q$  is read, " $P$  implies  $Q$ ", and indicates that you can find a proof of  $Q$  from hypothesis  $P$ . If you can, then  $P \Rightarrow Q$  must certainly be true according to the truth table, else  $P$  would be false and  $Q$  true, and that situation would invalidate the proof. But the mere fact that  $P \Rightarrow Q$  is true according to the table (i.e.  $P$  is false or  $Q$  true) doesn't have anything directly to do with your ability to construct a proof. Thus, truth tables are sometimes inappropriate for analyzing conditional sentences.

**Exercise 1** How many Boolean connectives are there? That is, how many truth tables are there? How many really depend on both variables  $P$  and  $Q$ ? For those that do, find corresponding English connective phrases.

You can easily extend the truth table method to handle sentences with more than two components. Here are two examples:

$P$	$Q$	$R$	$P \& (Q \vee R) \Leftrightarrow (P \& Q) \vee (P \& R)$				$P$	$Q$	$R$	$P \& (Q \vee R) \Leftrightarrow (P \& Q) \vee R$			
T	T	T	T	T	T	T	T	T	T	T	T	T	T
T	T	F	T	T	T	F	T	T	F	T	T	T	T
T	F	T	T	T	F	T	T	F	T	T	F	T	T
T	F	F	F	F	T	F	T	F	F	F	T	F	F
F	T	T	F	T	F	F	F	T	T	F	F	F	T
F	T	F	F	T	F	F	F	T	F	T	T	F	F
F	F	T	F	T	F	F	F	F	T	F	F	F	T
F	F	F	F	F	T	F	F	F	F	F	T	F	F

The sentence on the left,  $P \& (Q \vee R) \Leftrightarrow (P \& Q) \vee (P \& R)$ , is called a *tautology*: it's true independently of the truth values of its components. (In fact, they haven't even been specified!) The one on the right,  $P \& (Q \vee R) \Leftrightarrow (P \& Q) \vee R$ , is called a *contingency* because it's sometimes true, sometimes false.

**Exercise 2** Invent some sentences involving three (unspecified) components  $P$ ,  $Q$ ,  $R$  that are *false* independently of the truth values of  $P$ ,  $Q$ , and  $R$ . Such sentences are called *contradictions*.

The truth table method shown above on the left for verifying a tautology is called a *full sweep*. Sometimes you can short-cut it considerably by a *fell swoop*. For example, to verify that the sentence

$$((P \Rightarrow R)) \& ((Q \Rightarrow R)) \Rightarrow ((P \vee Q) \Rightarrow R)$$

is a tautology, attempt to falsify it. The left hand side would have to be true and the right hand false, so  $P \Rightarrow R$ ,  $Q \Rightarrow R$ , and  $P \vee Q$  would be true and  $R$  false. But falsehood of  $R$  would entail falsehood of  $P$  and  $Q$ , contradicting the truth of  $P \vee Q$ .

**Exercises 3** By one method or another verify the following tautologies. Most have names and play significant roles in applications of Boolean logic. For each tautology involving  $\&$  and  $\vee$ , construct a new one by interchanging those two connectives and possibly making other systematic changes (try reversing conditionals).

$(P \& P) \Leftrightarrow P$	Idempotency
$(P \& Q) \Leftrightarrow (Q \& P)$	Commutativity
$(P \& (Q \& R)) \Leftrightarrow ((P \& Q) \& R)$	Associativity
$(P \& (Q \vee R)) \Leftrightarrow ((P \& Q) \vee (P \& R))$	Distributivity
$(P \& Q) \Rightarrow P$	Lower bound
$(P \Rightarrow Q) \Rightarrow ((P \Rightarrow R) \Rightarrow (P \Rightarrow (Q \& R)))$	Greatest lower bound
$P \Rightarrow P$	Reflexivity
$(P \Rightarrow Q) \Rightarrow ((Q \Rightarrow P) \Rightarrow (P \Leftrightarrow Q))$	Weak asymmetry
$(P \Rightarrow Q) \Rightarrow ((Q \Rightarrow R) \Rightarrow (P \Rightarrow R))$	Transitivity
$(P \Rightarrow Q) \Leftrightarrow (\neg Q \Rightarrow \neg P)$	Contraposition
$\neg(P \& Q) \Leftrightarrow (\neg P \vee \neg Q)$	De Morgan's law
$\neg(P \& \neg P)$	Excluded middle
$\neg\neg P \Leftrightarrow P$	Double negation
$P \Rightarrow (Q \Rightarrow P)$	(No common name)
$(P \Rightarrow (Q \Rightarrow R)) \Rightarrow ((P \Rightarrow Q) \Rightarrow (P \Rightarrow R))$	Self distributivity
$((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$	Peirce's law

Although ancient philosophers studied some aspects of Boolean logic, the subject really stems from George Boole's work published in 1847. He noted that if you make some symbol substitutions in the previous list of formulas, the results often closely resemble valid algebraic laws. Here are some examples:

Replace	by	
$\&$	$\cdot$	
$\vee$	$+$	
$\neg$	$-$	
$\Leftrightarrow$	$=$	(sometimes)
$\Rightarrow$	$\leq$	(sometimes)
$P \cdot Q = Q \cdot P$		Commutativity
$P \cdot (Q \cdot R) = (P \cdot Q) \cdot R$		Associativity
$P \cdot (Q + R) = P \cdot Q + P \cdot R$		Distributivity
$P \leq P$		Reflexivity
$P \leq Q \Rightarrow (Q \leq P \Rightarrow P = Q)$		Weak asymmetry
$P \leq Q \Rightarrow (Q \leq R \Rightarrow P \leq R)$		Transitivity
$P \leq Q \Leftrightarrow \neg Q \leq \neg P$		Contraposition

$$\neg\neg P = P$$

Double negation

Of course, not all tautologies translate this way into valid algebraic laws. Boole's work suggested that algebraic methods might yield interesting results in logic, and during the next hundred years many mathematicians and philosophers contributed to that study. The exact relationship between truth tables and algebra is now contained in the highly developed and closely related subjects of Boolean logic and Boolean algebra. The field of algebraic logic extends these results to more detailed and powerful parts of logic.

Besides helping you verify tautologies, truth tables lead to *disjunctive normal forms* of sentences. If you substitute letters  $P, Q, \dots$  for the Boolean components of a non-contradictory sentence  $S$  as earlier, then you can construct a sentence  $D$  equivalent to  $S$ , which has a special form:  $D$  is a disjunction of conjunctions consisting of one each of the letters  $P, Q, \dots$  or their negations. *Equivalent* means that  $S \leftrightarrow D$  is a tautology. For example, reconsider the truth table of the example contingency  $S$  considered earlier:

$P$	$Q$	$R$	$S: P \& (Q \vee R) \leftrightarrow (P \& Q) \vee R$			
T	T	T	T	T	T	T
T	T	F	T	T	T	T
T	F	T	T	T	F	T
T	F	F	F	F	T	F
F	T	T	F	T	F	T
F	T	F	F	T	T	F
F	F	T	F	T	F	T
F	F	F	F	F	T	F

The T entries in the third column from the right say that  $S$  is true just in case

$P$  is true,  $Q$  is true, and  $R$  is true, or  
 $P$  is true,  $Q$  is true, and  $R$  is false, or  
 $P$  is true,  $Q$  is false, and  $R$  is true, or  
 $P$  is true,  $Q$  is false, and  $R$  is false, or  
 $P$  is false,  $Q$  is true, and  $R$  is false, or  
 $P$  is false,  $Q$  is false, and  $R$  is false.

That is,  $S$  is equivalent to the following sentence  $D$ , because they have the same truth table:

$$(P \& Q \& R) \vee (P \& Q \& \neg R) \vee (P \& \neg Q \& R) \vee (P \& \neg Q \& \neg R) \vee (\neg P \& Q \& \neg R) \vee (\neg P \& \neg Q \& \neg R).$$

To design an electrical circuit with input signals represented by  $P, Q, \dots$  whose output signal behaves like a given Boolean combination  $S$  of  $P, Q, \dots$  engineers can

select electronic components called *gates* that affect the output like the connectives  $\&$ ,  $\vee$ , and  $\neg$ , then wire them together guided by the normal form  $D$ . To economize on gates, they want a version of  $D$  with as few connectives as possible. This leads to a subject called *Boolean minimization*. For example, the disjunction of the first four conjuncts of the normal form  $D$  just considered is equivalent to the simple sentence  $P$ . Thus, the original sentence  $S$  is equivalent to the shorter form

$$P \vee (\neg P \& Q \& \neg R) \vee (\neg P \& \neg Q \& \neg R).$$

**Exercise 4** Find a sentence equivalent to this example that includes only  $P, Q, R$ , the connectives  $\&$ ,  $\vee$ ,  $\neg$ , and parentheses, and has as few connectives as possible.

**Exercise 5** Show that every non-tautological sentence with Boolean components  $P, Q, \dots$  is equivalent to a conjunction  $C$  of disjunctions that contain one each of the components and their negations. (Hint: use De Morgan's law.) Find this *conjunctive normal form*  $C$  for the example sentence  $S$  just considered.

Another way to economize in circuit design is to use gates of one type only. This is possible because of another normal form result: every sentence  $S$  is equivalent to a sentence  $N$  in which no Boolean connective occurs except  $\downarrow$  (neither/nor). To construct  $N$  first find a sentence  $C$  equivalent to  $S$  with no connectives except  $\&$ ,  $\vee$  and  $\neg$ . Second, replace all conjunctions  $T \& U$  in  $C$  by equivalent formulas  $\neg T \downarrow \neg U$ . You get a new sentence  $C'$  equivalent to  $S$  with no connectives except  $\downarrow$ ,  $\vee$ , and  $\neg$ . Third, replace all disjunctions  $T \vee U$  in  $C'$  by equivalent formulas  $\neg(T \downarrow U)$ , and delete any double negatives  $\neg\neg$ . You get a new sentence  $C''$  equivalent to  $S$  with no connectives except  $\neg$  and  $\downarrow$ . Finally, form the desired sentence  $N$  by replacing all negations  $\neg T$  in  $C''$  by equivalent formulas  $T \downarrow T$ . To apply this result to circuit design, convert the output signal sentence  $S$  to normal form  $N$ , select *nor* gates that affect the output signal like the connective  $\downarrow$ , and wire them together according to  $N$ .

**Exercise 6** Construct a normal form of type  $N$  equivalent to the example sentence  $S$  considered earlier. Try to make  $N$  as short as possible.

**Exercise 7** Show that every sentence  $S$  is equivalent to a sentence  $M$  in which no Boolean connective occurs except the one (called *Sheffer's stroke*, or *nand*) with truth table

$P$	$Q$	$P \uparrow Q$
T	T	F
T	F	T
F	T	T
F	F	T

Construct such a normal form  $M$  equivalent to the example sentence  $S$  considered earlier. Try to make  $M$  as short as possible.