St. Pölten University of Applied Sciences

/informatik & security

/fh/// st. pölten

Truth Tables

Theoretical Computer Science

Dipl.-Ing. Hubert Schölnast, BSc.

October 03, 2021

Table of Contents

1 Truth table for just one variable p

f = false
t = true

in	out			
p	$\bot$	p	$\neg p$	$\top$
f	f	f	t	t
t	f	t	f	t

0 = false
1 = true

in	out			
p	$\bot$	p	$\neg p$	$\top$
0	0	0	1	1
1	0	1	0	1

There are 4 possibilities for the output:

$\bot$ "Contradiction" is always false. Since it is independent from any input, it normally will be written without any parameter. (You don't have to write "$\bot\ p$" just "$\bot$" is enough because the contradiction is a 0-ary or constant function.)

p This just is p itself, without any operators.

$\neg p$ "Not p". This is the negation of p.

$\top$ "Tautology" is always true. And like the contradiction normally written without a parameter.

Because contradiction and tautology do not depend on p, and because the output p is just a copy the input, of these 4 possibilities only $\neg p$ is really interesting. You just should know, that for 1 input variable, you can define 4 different truth functions.

2 Truth table for no input variables at all

Since contradiction and tautology don't need any input at all, you also can create a truth table for no variables at all.

in	out	
	$\bot$	$\top$
	f	t

in	out	
	$\bot$	$\top$
	0	1

As you can see, for 0 input variables, you can define 2 different functions.

3 Truth table for two input variables p and q

Even more interesting are the binary operators. They are called “binary” (Latin *bis* = double, *bini* = twofold, *binarius* = consists of two) because they combine two input variables which here shall be p and q. (“Binary numbers” have their name also from the same Latin root, because they are written with just two different digits, 0 and 1, but still binary operators are not defined as operators for binary numbers. Binary operators are defined as operators which take two variables as input.)

in		out															
p	q	$\bot$	$p \wedge q$	$p \nrightarrow q$	p	$p \nleftarrow q$	q	$p \nleftrightarrow q$	$p \vee q$	$\neg(p \vee q)$	$p \leftrightarrow q$	$\neg q$	$p \leftarrow q$	$\neg p$	$p \rightarrow q$	$\neg(p \wedge q)$	$\top$
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

From these 16 possibilities 2 don’t depend on any input ($\bot$ and $\top$), 4 of them depend on only one of the two input parameters ($p, q, \neg p$ and $\neg q$) and the remaining 10 are “real” binary operators.

Only 2 of them are really fundamental:

$p \wedge q$ “p and q” is only true if all input parameters are true. “p and q” is false if at least one of its input variables is false. Other very common symbols are the dot $p \cdot q$ and the ampersand $p \,\&\, q$.

$p \vee q$ “p or q” is already true if at least one of its input parameters is true. It only will become false if all input parameters are false. Other very common symbols are the plus sign $p + q$ and the vertical bar $p \mid q$.

Two others are just negations of “and” and “or”:

$\neg(p \wedge q)$ This operator is called “NAND” (Not + AND) and is the negation of the result of an “and” operation. There also exists the symbol $p \uparrow q$, but it is used very rarely.

$\neg(p \vee q)$ “NOR” (Not + OR) (In rare cases written as $p \downarrow q$.)

Three more operators are functions with their own symbols, but they can also be written as combinations of $\wedge$, $\vee$ and $\neg$:

$p \rightarrow q$ “Implication”. See section 3.1 for more details.
One way to write $p \rightarrow q$ as a combination of of “$\wedge$”, “$\vee$” and “$\neg$” is “$\neg(p \wedge \neg q)$”.
Another one is “$\neg p \vee q$”.

$p \leftarrow q$ This is the converse implication. It can be written as “$p \vee \neg q$” or “$\neg(\neg p \wedge q)$”.

$p \leftrightarrow q$ This has many names. Commonly used names are “biconditional” but also “equality”. This function is true if p and q have the same truth values. If p and q have different truth values, $p \leftrightarrow q$ is false.
It can be written as “$(p \rightarrow q) \wedge (p \leftarrow q)$” which is equivalent to “$(p \wedge q) \vee (\neg p \wedge \neg q)$”

The remaining three functions are negations of the previous three:

$p \nrightarrow q$ "Negated implication". It is the negation of the implication: $\neg(p \rightarrow q)$.
As a combination of of "∧", "∨" and "¬" this is "$p \wedge \neg q$" or "$\neg(\neg p \vee q)$".

$p \nleftarrow q$ "Negated converse implication" $\neg(p \leftarrow q)$.
"$\neg p \wedge q$" or "$\neg(p \vee \neg q)$".

$p \nleftrightarrow q$ "Exclusive Or". It can be defined as the negation of equality: $\neg(p \leftrightarrow q)$. But more often it is defined as that version of an "or"-Operation, that is true if either p or q is true, but not when both of them are true (in which case $p \nleftrightarrow q$ is false). And because of this close relation to the "normal" OR, this function is also well known as XOR and there also is a very common alternate symbol: $p \oplus q$.

3.1 Implication

The function $p \rightarrow q$ needs some explanation:

"p implies q" or "if p then q" means: If p is true, then the expression $p \rightarrow q$ will inherit its value from q. But if p is false, then the expression will be true. This may seem counterintuitive at first sight, so let's make it clear with an example:

Let $p =$ "It is raining" and $q =$ "Streets are wet".

Let's first see, what happens if $p = t$ (true) (It is raining):

- $p = t,\ q = t$: "It is raining, and the streets are wet."
  This is true: $(t \rightarrow t) = t$
- $p = t,\ q = f$: "It is raining, and the streets are not wet (streets are dry)."
  This is not possible. Whenever it rains, you will see that the streets will become wet. So, the statement *"It is raining, and the streets aren't wet"* cannot be true. It is false:
  $(t \rightarrow f) = f$

This was easy, so now have a closer look to a sunny and warm day with dry weather:

- $p = f,\ q = f$: "It is not raining, and the streets aren't wet."
  This is something, that you can observe almost every day: On a sunny and warm day with dry weather also the streets are dry. So, the statement "It is not raining, and the streets aren't wet" is true: $(f \rightarrow f) = t$
- $p = f,\ q = t$: "It is not raining, and the streets are wet." Is it possible, that the streets are wet although the weather is dry? Yes of course, this is possible. In many towns and villages, the fire brigade spills water onto the streets on really hot days. And then, even if there is no cloud up in the sky, the streets still are wet. So, a wet street is a possible implication of dry weather, and this means: $(f \rightarrow t) = t$.

4 Size of Truth Tables

No logic variables at all:

in	out	
	$\bot$	$\top$
	0	1

1 row, 2 columns

1 logic variable:

in	out			
p	$\bot$	p	$\neg p$	$\top$
0	0	0	1	1
1	0	1	0	1

2 rows, 4 columns

2 logic variables:

in		out															
p	q	$\bot$	$p \wedge q$	$p \nrightarrow q$	p	$p \nleftarrow q$	q	$p \nleftrightarrow q$	$p \vee q$	$\neg(p \vee q)$	$p \leftrightarrow q$	$\neg q$	$p \leftarrow q$	$\neg p$	$p \rightarrow q$	$\neg(p \wedge q)$	$\top$
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

4 rows, 16 columns

3 logic variables:

in			out						
a_0	a_1	a_2	f_0	f_1	f_2	$\cdots$	f_{253}	f_{254}	f_{255}
0	0	0	0	0	0	$\cdots$	1	1	1
0	0	1	0	0	0	$\cdots$	1	1	1
0	1	0	0	0	0	$\cdots$	1	1	1
0	1	1	0	0	0	$\cdots$	1	1	1
1	0	0	0	0	0	$\cdots$	1	1	1
1	0	1	0	0	0	$\cdots$	1	1	1
1	1	0	0	0	1	$\cdots$	0	1	1
1	1	1	0	1	0	$\cdots$	1	0	1

8 rows, 256 columns

4.1 Any number of logic variables

Let's summarize the results:

variables	rows	columns
0	1	2
1	2	4
2	4	16
3	8	256

The pattern is very obvious:

variables	rows	columns
0	1	2
1	2	4
2	4	16
3	8	256
n	2^n	2^{2^n}

Any column in a truth table is a distinct logical formula. Every logical formula can be written as combinations of logical variables and these three operators:

- not: $\neg p$
- or: $(p \vee q)$
- and: $(p \wedge q)$