MAT120: Lecture 3 Handout

Formal Logic

It would be nice if there were a computer program that we could use to feed in arguments and spit out some sort of "yes or no" answer that tells us whether the argument is valid or not. As a matter of fact, this is actually possible, and it is called your brain! In this lecture we will present the simplest version of this computer program: so-called *truth tables*. What we will do is take the ideas and concepts from last lecture and attach symbols to everything, to see how we can "compute" the arguments to check if they are valid or not.

1 Logic

1.1 A Motivation

Consider the following three examples:

- "If it rained last night then the ground would be wet this morning. But the ground is not wet this morning. Therefore, it could not have rained last night."
- "If Jimmy drinks coffee then he gets very excited. But he seems quite calm right now, so he cannot have had a coffee."
- "If my package was delivered then there would be a delivery note in my mailbox. There is no note in my mailbox, so my package must not have been delivered."

These three statements are talking about different things. But at the same time, they are somehow the same. They are all saying something like "if p then q. But not q, so therefore not p".

As a matter of fact, they are all examples of an underlying law of Logic! This particular law is called *Modus Tollens*. Lots of these laws of logic have fancy Latin or Greek names, because they were discovered in Ancient Times by people like Aristotle (and others). The statement of Modus Tollens is below.

Modus Tollens

From "p implies q" and "not q" we can deduce "not p".

There are all sorts of interesting logical laws. In this course we will keep things simple and only present three of them. However, it is worth noting that you already know all of the other laws of logic. You use them every day when you think about things and navigate the world—logic and reasoning are fundamental to our species, and you cannot help but reason about the world when you interact with it. Here are three particular laws of logic that we will care about in this course:

- **Modus Tollens:** From "p implies q" and "not q" we can deduce "not p".
- **Modus Ponens:** From "p implies q" and "p" we can deduce "q".
- **Hypothetical Syllogism:** From "p implies q" and "q implies r", we can deduce "p implies r".

Now, with this in mind, what is "Logic"? In fact, the best way to see it is that Logic is a language! Treating Logic as a language allows us to:

- Express logical statements *generally* without needing to use specific examples.
- Describe the "structure" of the reasoning being used.
- Check whether reasoning is valid or not.

1.2 What is a Language?

Before presenting the details, we must first ask: if Logic is a language, then what on earth is a language? The exact definition and features of language are a bit complicated, but very deep and informative. We will cut the discussion down to the important truth: any language is a symbolic representation of something meaningful. This means that a language is a "collection of symbols" with:

- a bunch of rules for how to combine the symbols into new "good" combinations;
- an interpretation or meaning attached to the symbols.

This is a little bit abstract and confusing, so let's take English as an example. In English, we have a collection of symbols: a, b, c, d, ..., as well as punctuation marks like brackets, apostrophes, quotation marks, and so on. Whether you have noticed it or not, there is a big list of rules for how to combine these symbols to make new "good" ones. For example: we know that in English "tree" is a word, since it combines the symbols t, r, e, and e in the right way. However, we also know that "eert" is not a word, since it combines the symbols t, r, e, and e in the wrong way. This covers the first point above. Alongside knowing that something is a good combination of symbols, we also have an interpretation of the symbols: "tree" also means something (these symbols mean the big wooden thing in the park or the forest).

Let's pick another example: in Japanese, we have a collection of symbols: hiragana, katakana, and kanji. We know that these symbols can be combined to make new words, but only in the "right" way. Again, there is an interpretation at play: the symbol き has the same interpretation as the English word "tree". In fact, this is precisely what translation between languages is doing: we are exchanging symbols in one language into symbols in another language, without changing the interpretation of the symbols. Interestingly, in Japanese, the idea of interpretation is even more clear: in Japanese there are multiple distinct symbols that have the same interpretation. This is the difference between き and 木, for example.

Examples such as English or Japanese occur naturally in the real world. For that reason, we call them *natural* languages. Logic is a different style of language: it is artificial—made up by clever people like Aristotle, in order to better understand the laws of reasoning.

1.3 What does Logic look like?

Keeping the two ingredients of language in mind (symbols, and an interpretation), in order to define "Logic", we need to specify a collection of symbols and an interpretation of them. Here are the details:

- Symbols p, q, r, ... that represent logical statements.
- Symbols that represent the logical connectives, listed below:
  - Negation: $\neg$
  - Conjunction: $\wedge$
  - Disjunction: $\vee$
  - Implication: $\rightarrow$
  - Equivalence: $\leftrightarrow$
- Symbols (and), which are regular brackets.

A *formula* is any expression that can be built from the symbols $p, q, r, \cdots$ using the connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ and brackets, following the expected rules. For instance, formulas may look like: p, $\neg q$, $(p \vee r) \rightarrow q$ and so on, whereas combinations like $p\wedge \rightarrow r\vee$ or $\vee pq\neg r$ are not formulas.

This language is called *Propositional Logic*, because the word "proposition" is just another term for a logical statement. So, this language of symbols that I have listed above is simply the language of logical statements (in fact, it is the simplest version of such a language).

In natural languages like English or Japanese, the symbols that we use have an interpretation, i.e.

the symbols on a page are referring to a real-world object or concept. In the case of propositional logic, the "real-world objects" don't really exist anymore. Instead, the symbols p, q, r, $\dots$ have one of two possible interpretations: they are either true (T) or false (F). That is why we only consider logical statements for p, q, r, $\dots$—we are interested in studying their possible truth values.

Using this propositional logic, we can translate sentences into mathematics:

- "not p" becomes $\neg p$
- "p and q" becomes $p \wedge q$
- "p or q" becomes $p \vee q$
- "If p then q" becomes $p \to q$
- "p if and only if q" becomes $p \leftrightarrow q$

The brackets (and) are needed to tell us the order in which to evaluate things: we do the operations in the brackets first, and then we look at the operations outside the brackets.

1.4 Example

In order to see how to use all of these symbols, we will now practice translating normal English into the language of propositional logic.

Let p = "I own a dog" and q = "I own a cat". I will translate the following statements into propositional logic:

- If I own a dog, then I own a cat.
  A: $p \to q$
- If I own a cat, then I own a dog.
  A: $q \to p$
- I own a dog and a cat.
  A: $p \wedge q$
- I don't own a dog, but I own a cat.
  A: $(\neg p) \wedge q$

In the final example, we write the brackets because writing $\neg(p \wedge q)$ means something different (i.e. it represents a different argument).

1.5 Exercise

Exercise

Let $p =$ "I am wearing sandals" and $q =$ "I am at the beach". Convert the following statements into propositional logic:

- I am at the beach or I am wearing sandals.
  A:
- If I am at the beach then I am wearing sandals.
  A:
- I am not at the beach and I am not wearing sandals.
  A:
- I am at the beach if and only if I am wearing sandals.
  A:

1.6 Expressing Modus Tollens in Propositional Logic

Going back to Modus Tollens for a moment, recall the argument below:

> "If it rained last night then the ground would be wet this morning. But the ground is not wet this morning. Therefore, it could not have rained last night."

If we let $p =$ "it rained last night" and $q =$ "the ground is wet this morning", then the argument can be rephrased by saying something like "we can use the two statements $(p \to q)$ and $\neg q$ to deduce that $\neg p$ also holds". To be more accurate, in symbols we have that the formula

$$\big((p \to q) \land \neg q\big) \to \neg p$$

is always true no matter what p and q mean and whether or not they are true. In particular, this general "law of logic" also applies to the above argument about rainwater. Cool, huh!

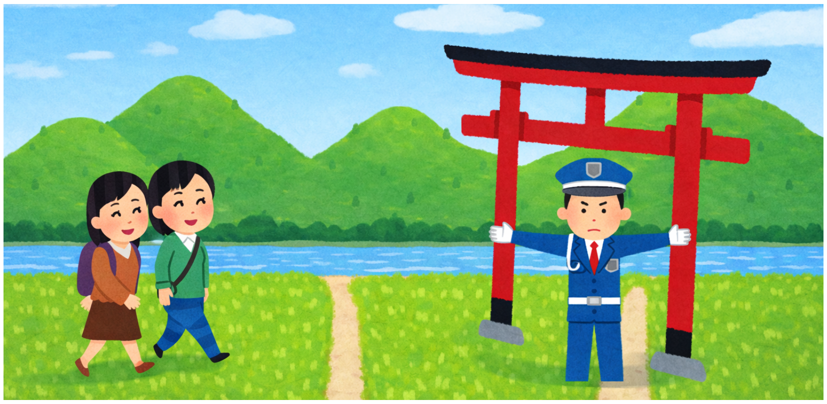
2 Truth Tables

We will now discuss the "computer program" mentioned at the start of this handout. In short, there is indeed an algorithm that one can employ in order to check whether an argument is valid or not. This algorithm takes a formula written in propositional logic, and evaluates it by running through all the possible truth-values of its variables $p, q, r, \ldots$ and checking if these make the overall formula true or false. This entire "checking" procedure can be displayed all at once in something called a *truth table*, which we will now summarize. We start with a nice analogy.

2.1 An Analogy using Torii Gates

Let's pretend that you and your friend are going on a walk through the countryside.

2.1.1 The Painted Torii



On your walk, you discover a torii gate that is painted red. At the gate there is a strange man standing there. He says to you "Stop! I am the guardian of this gate. I will not let you through unless you pass my test." You and your friend reply "ok, sure!". The man then says

> "The road ahead is dangerous. I will not let you pass through the gate unless you have a map and a compass."

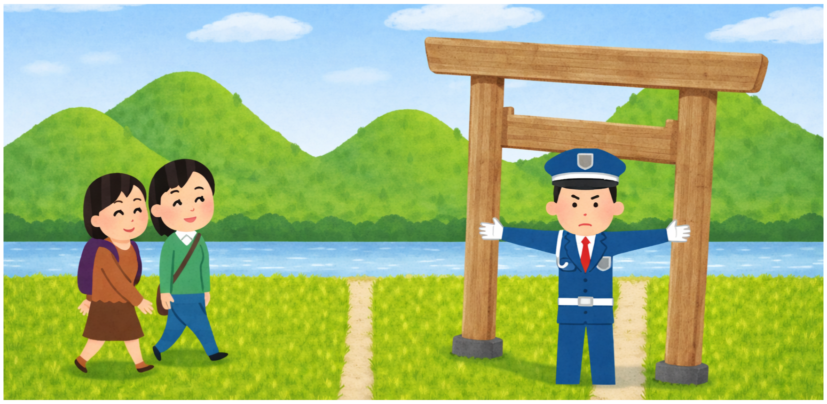
You do not remember if you have a map and compass in your bags, so you need to check. Before you check in your bags, let's think about the possible options. There are only four possible ways that things can turn out:

Case 1: you do not have a map and you do not have a compass;
Case 2: you do not have a map, but you do have a compass;
Case 3: you have a map, but you do not have a compass;
Case 4: you have both a map and a compass.

According to what the strange man said, in order to pass his test, you need to show him both a map and a compass. So, in the four possible cases, there is only one situation where you can pass the test, which is Case 4.

Fortunately, you check your bags, and success! You happen to find a map and a compass. So, the strange man lets you pass through, and you continue your walk.

2.1.2 The Wooden Torii



You and your friend continue your nice countryside walk. On your walk you find a second torii gate. This gate is different from the first – now it is made out of wood, so it looks brown and organic. Amazingly, you find another strange man standing at the gate! This man says: "Stop! I am the guardian of this gate. I will not let you through unless you pass my test." Again, you and your friend reply "ok, sure!". This strange man says

> "It is getting dark soon. I will only let you pass through if you have a phone or a flashlight (or both)."

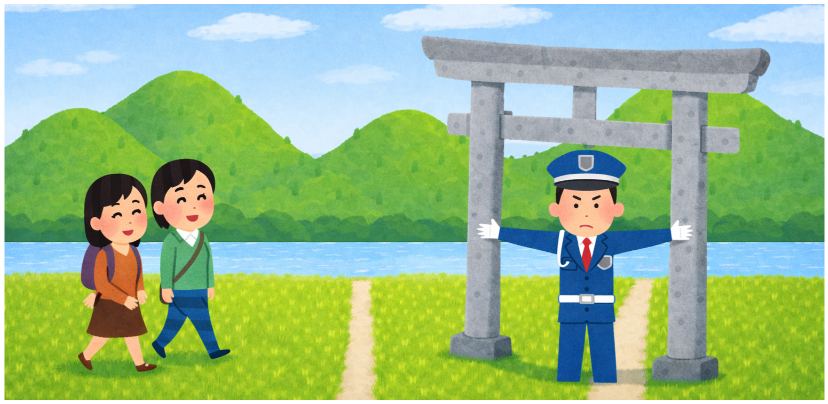
You again say "ummm" and start to check your bags. Again, following the same reasoning from before, there are only four possible situations:

Case 1: you do not have a phone and you do not have a flashlight;
Case 2: you do not have a phone, but you do have a flashlight;
Case 3: you have a phone, but you do not have a flashlight;
Case 4: you have both a phone and a flashlight.

According to what he said, this strange man requires that you have either a phone, or a flashlight, or both. Therefore, there are now three possible situations in which you will pass his test: Cases 2, 3, and 4 will all be ok. He will only prevent you from passing through the gate in Case 1, i.e. when you don't have a phone or a flashlight.

Anyway, you check your bags and you find a flashlight. So, the strange man says "great!" and lets you pass.

2.1.3 The Stone Torii



You and your friend keep walking, and you find a third torii gate! This one is different from the other two: instead of being painted or being wooden, this third torii gate is made from stone. So, it looks tough and unforgiving. As you may expect, there is a third strange man at the stone torii. He says to you "Stop! I am the guardian of this gate. I will not let you through unless you pass my test." At this point, you and your friend are starting to find this quite funny. Anyway, you both reply "ok, sure!", and the man says:

> "You are about to enter a wildlife protection zone. If you are carrying a camera, then you need to provide a camera permit in order to enter."

Again, following the same reasoning from before, there are only four possible cases:

Case 1: you do not have a camera and you do not have a camera permit;
Case 2: you do not have a camera, but you do have a camera permit;
Case 3: you have a camera, but you do not have a camera permit;
Case 4: you have both a camera and a camera permit.

According to the rules of the third strange man, he will only stop you if he finds that you have a camera but no camera permit. In Cases 1 and 2 you don't have a camera, so the strange man doesn't care about whether or not you have a camera permit. In Case 4, you have both a camera and a camera permit, so everything is fine. However, in Case 3 you explicitly violate the strange man's rule: you have a camera but you do not have a permit. So, Case 3 is the only situation in which you will fail his test.

2.2 Defining a Truth Table

By definition, logical statements always have a truth-value, either true or false. So, if we pick two of these statements, p and q, there are only four possible ways for things to be:

- Case 1: p is false and q is false
- Case 2: p is false and q is true
- Case 3: p is true and q is false
- Case 4: p is true and q is true

Notice the similarity here between these four cases and the examples from before: it's basically the same thing! Instead of writing out "Case 1, Case 2, Case 3, Case 4" like we did before, this time we write everything out in a table that displays all of the information in a neat visual package. We let "T" mean "True" and "F" mean "False". A "truth table" is just a way of writing out all the possible cases into a table, like below.

Case	p	q
1	F	F
2	F	T
3	T	F
4	T	T

2.3 Truth Tables for the Connectives

The three binary connectives $\wedge$, $\vee$ and $\rightarrow$ are the three torii gates from before! I will write them out for you one-by-one.

2.3.1 Conjunction

Conjunction has a truth table that looks like the painted torii gate. The truth table for the conjunction $p \wedge q$ is written by adding an extra column to the one from before:

Case	Map?	Compass?	
1	No	No	No
2	No	Yes	No
3	Yes	No	No
4	Yes	Yes	Yes

"No"=F $\longrightarrow$ "Yes"=T

Case	p	q	$p \wedge q$
1	F	F	F
2	F	T	F
3	T	F	F
4	T	T	T

If we let p = "We have a map" and q = "We have a compass", then the four cases from our example directly map to this truth table: the "passing of the strange man's test" corresponds to an output of "True" in the truth table, and the "failing of the strange man's test" corresponds to an output of "False". The only time that a conjunction can be true is when both statements are true.

2.3.2 Disjunction

Disjunction has a truth table that looks like the wooden torii gate. The truth table is written below.

Case	Phone?	Flashlight?	
1	No	No	No
2	No	Yes	Yes
3	Yes	No	Yes
4	Yes	Yes	Yes

"No"=F → "Yes"=T

Case	p	q	$p \vee q$
1	F	F	F
2	F	T	T
3	T	F	T
4	T	T	T

The table is telling us that there are three possible ways that a disjunction can be true: if either of the two statements is true (or both). In other words, the only way that a disjunction is false is if both statements are false at the same time.

2.3.3 Implication

Implication has a truth table that looks like the stone torii gate. It is written below.

Case	Camera?	Permit?	
1	No	No	Yes
2	No	Yes	Yes
3	Yes	No	No
4	Yes	Yes	Yes

"No"=F → "Yes"=T

Case	p	q	$p \to q$
1	F	F	T
2	F	T	T
3	T	F	F
4	T	T	T

This situation is just like the stone torii gate from before: $p =$ "we have a camera" and $q =$ "We have a camera permit". Then, there is only one situation in which the implication fails, i.e. p is true and q is false (which was represented by saying that we had a camera but not a permit).

2.4 Computing Truth Tables for Other Formulas

As mentioned previously, we can use the logical connectives to build more complicated formulas from smaller ones. The key to evaluating the truth-value of these complicated formulas is to break them down into smaller pieces that we know how to do.

As an example of this, we will consider the *Modus Tollens* formula mentioned previously: $\big((p \to q) \wedge \neg q\big) \to \neg p$. In order to write the truth table for the formula $\big((p \to q) \wedge \neg q\big) \to \neg p$, the first thing that we must do is use the brackets (and) to "break down" the formula into its smaller pieces. As we can see from the formula, there are four smaller pieces of the formula:

- Piece 1: the formula $(p \to q)$
- Piece 2: the formula $\neg q$
- Piece 3: the formula $(p \to q) \wedge \neg q$
- Piece 4: the formula $\neg p$

These combine to give the final formula $\big((p \to q) \wedge \neg q\big) \to \neg p$. To create the truth table for this formula, we first write all of these subformulas as different columns. In order to better explain the technique, the columns are numbered.

		(C1)	(C2)	(C3)	(C4)	(C5)
p	q	$(p \to q)$	$\neg q$	$(p \to q) \wedge \neg q$	$\neg p$	$\big((p \to q) \wedge \neg q\big) \to \neg p$
F	F					
F	T					
T	F					
T	T					

Now from here, we fill in each column by using the truth tables for the connectives $\wedge$, $\vee$, $\to$ and the negation $\neg$, which simply inverts the truth-value of its input. Starting from the left, we fill in each column one-by-one:

- **Column (C1):** we are trying to evaluate an implication, i.e. $p{\to}q$. According to the truth table for implication (the stone torii gate of Section 2.1.3 and 2.3.3), the implication $p \to q$ will only be false when p is true and q is false. So, column (C1) will read "T T F T":

		(C1)	(C2)	(C3)	(C4)	(C5)
p	q	$(p \to q)$	$\neg q$	$(p \to q) \wedge \neg q$	$\neg p$	$\big((p \to q) \wedge \neg q\big) \to \neg p$
F	F	T				
F	T	T				
T	F	F				
T	T	T				

- **Column (C2):** we are trying to evaluate a negation, i.e. $\neg q$. Negations simply flip the truth-value of the input. In this case, the column for q says that the truth-values for q are "F T F T". So the negation just inverts these, which make the columns "T F T F":

		(C1)	(C2)	(C3)	(C4)	(C5)
p	q	$(p \to q)$	$\neg q$	$(p \to q) \wedge \neg q$	$\neg p$	$\big((p \to q) \wedge \neg q\big) \to \neg p$
F	F	T	T			
F	T	T	F			
T	F	F	T			
T	T	T	F			

- **Column (C3):** we are trying to evaluate a conjunction, i.e. $(p \to q){\wedge}\neg q$. Conjunctions are only true when both inputs are true. In this case, that means that we need both $p \to q$ and $\neg q$ to be true. Columns (C1) and (C2) contain all of the truth-values of $p \to q$ and $\neg q$, respectively. So, we simply look through (C1) and (C2) and try to find a row in which both are Ts. Upon inspection, we see that Row 1 has both $p \to q$ and $\neg q$ being true, and Rows 2, 3 and 4 have at least one of these two statements being false. Therefore, Row 1 is the only situation which will make the conjunction $(p \to q) \wedge \neg q$ true. In every other case, it will be false. We write out this column:

		(C1)	(C2)	(C3)	(C4)	(C5)
p	q	$(p \to q)$	$\neg q$	$(p \to q) \wedge \neg q$	$\neg p$	$\big((p \to q) \wedge \neg q\big) \to \neg p$
F	F	T	T	T		
F	T	T	F	F		
T	F	F	T	F		
T	T	T	F	F		

- **Column (C4):** here, again we are taking a simple negation, i.e. $\neg p$. In this case, we have something similar to column (C2), except that here we are negating p instead of q. So, we look to the column for p and see that it reads "F F T T", and invert everything. The column (C4) will therefore read "T T F F":

		(C1)	(C2)	(C3)	(C4)	(C5)
p	q	$(p \to q)$	$\neg q$	$(p \to q) \wedge \neg q$	$\neg p$	$\big((p \to q) \wedge \neg q\big) \to \neg p$
F	F	T	T	T	T	
F	T	T	F	F	T	
T	F	F	T	F	F	
T	T	T	F	F	F	

- **Column (C5):** finally, we consider the implication $\big((p \to q) \wedge \neg q\big) \to \neg p$. Similarly to (C1), we use the stone torii example of Sections 2.1.3 and 2.3.3. Again, the only time that an implication will be false is when the first statement is true and the second statement is false. In the formula $\big((p \to q) \wedge \neg q\big) \to \neg p$ the first statement is $\big((p \to q) \wedge \neg q\big)$ and the second formula is $\neg p$. So, in order to evaluate the implication "$\to$" we need to look at the truth-values listed in columns (C3) and (C4). We are looking for a row in the table for which column (C3) is true yet column (C4) is false, because that will make the connective "$\to$" false also. However, in this case, we see that in the 4 rows, there are *no* situations in which (C3) has a T and (C4) has an F. So, there is no situation in which the implication "$\to$" will be false. Therefore, every entry is a T, and this completes the table:

p	q	$(p \to q)$	$\neg q$	$(p \to q) \wedge \neg q$	$\neg p$	$\big((p \to q) \wedge \neg q\big) \to \neg p$
F	F	T	T	T	T	T
F	T	T	F	F	T	T
T	F	F	T	F	F	T
T	T	T	F	F	F	T

For example, we can break the formula $(p \vee q) \to (p \wedge q)$ into smaller pieces, $p \vee q$ and $p \wedge q$, which are then connected with the implication $\to$. Therefore, in order to construct a truth table for this formula, we see that the implication comes *last*, and the two sub-formulas $p \vee q$ and $p \wedge q$ need to be evaluated first.

2.5 The Uses of Truth Tables

1. Truth tables can tell us whether a formula is a law of logic or not.
2. Truth tables can tell us which values of p and q cause a formula to be true (or false).
3. Truth tables can tell us whether two formulas are secretly the same (i.e. they mean the same thing).

We will now discuss these three topics separately. Along the way, I will also provide some examples that tell you how to compute a truth table for a given formula.

2.5.1 Identifying a Tautology

Consider the statement "If it rained last night then the ground would be wet this morning. But the ground is not wet this morning. Therefore, it could not have rained last night". Based on our discussions from earlier, we know that this is an example of Modus Tollens, which I already said was a logical law. In fact, these logical laws have a special name: they are called "tautologies", which is a fancy Greek word that means "saying the same again".

We can see the fact that Modus Tollens is a logical law by inspecting its truth table. In Section 2.4 we computed the truth table for Modus Tollens. Look again at the final column.

p	q	$(p \to q)$	$\neg q$	$(p \to q) \land \neg q$	$\neg p$	$\big((p \to q) \land \neg q\big) \to \neg p$
F	F	T	T	T	T	T
F	T	T	F	F	T	T
T	F	F	T	F	F	T
T	T	T	F	F	F	T

As you can see, every entry in the final column is true. This means that the formula $\big((p \to q) \land \neg q\big) \to \neg p$ is always true *no matter what*. In other words, it is a tautology. Generally speaking, a formula will be a tautology whenever the final column of its truth table has all Ts.

2.5.2 Identifying a Falsehood

Imagine a situation in which somebody gives bad reasoning: "If I have a dog, then I have an animal. I have an animal, therefore I have a dog", which has logical formula $\big((p \to q) \land q\big) \to p$. This argument is clearly bad, and we can demonstrate it using truth tables.

The completed truth table for this formula is written below. Please keep in mind the values in the final column.

p	q	$(p \to q)$	$(p \to q) \land q$	$\big((p \to q) \land q\big) \to p$
F	F	T	F	T
F	T	T	T	F
T	F	F	F	T
T	T	T	T	T

Notice that in the second row, the final column has an F there. This means that there is a situation in which the formula $\big((p \to q) \land q\big) \to p$ is false. Going back to the start of that row, we see that the formula will be false when p is false and q is true.
We can translate this back into English. The argument said: "If I have a dog, then I have an animal. I have an animal, therefore I have a dog", where $p =$ "I have a dog" and $q =$ "I have an animal". According to our truth table, the argument is false when q is true but p is false. Translated into English, this situation is saying that the overall argument is false whenever "I have an animal but I do not have a dog". But this makes perfect sense! Just because you have an animal does not mean that you have a dog. For example, you could have a cat and that would make q true and p false. So, we have successfully used a truth table to identify where this argument breaks down.

2.5.3 Identifying an Equivalence

For our third and final use of truth tables, we will test whether two formulas are actually the same. Consider the two formulas $\neg p \vee q$ and $p \to q$. I will now demonstrate that these mean the same thing, i.e. they have the same function on p and q. Let's look at the truth table for $\neg p \vee q$.

p	q	$\neg p$	$\neg p \vee q$
F	F	T	T
F	T	T	T
T	F	F	F
T	T	F	T

In the table above, $\neg p \vee q$ is true in Rows 1, 2 and 4, and is false in Row 3. However, that is exactly the truth table for the implication $p \to q$! Therefore, we can conclude that these two formulas are doing the same thing to p and q. In other words: truth tables have just shown us that the formulas $\neg p \vee q$ and $p \to q$ are actually equivalent (i.e. the same).

2.6 Exercise

Exercise

We are going to check if the two formulas $\neg(p \wedge q)$ and $(\neg p) \vee (\neg q)$ are equivalent or not. To do that, we need compute two truth tables: one for $\neg(p \wedge q)$ and one for $(\neg p) \vee (\neg q)$ and then compare them. Fill in the two truth tables below.
Here is the truth table for $\neg(p \wedge q)$ with entries in the gaps. Please delete the entries and replace them with T's or F's.

p	q	$p \wedge q$	$\neg(p \wedge q)$
F	F		
F	T		
T	F		
T	T		

p	q	$\neg p$	$\neg q$	$(\neg p) \vee (\neg q)$
F	F	T	T	T
F	T	T	F	T
T	F	F	T	T
T	T	F	F	F

Using your answers for the two truth tables, decide whether the formulas $\neg(p \wedge q)$ and $(\neg p) \vee (\neg q)$ are the same or not.
A:

2.7 A Further Explanation of How to Compute Truth Tables

As an addendum to Section 2.4, I will now give an extended explanation for how to compute truth tables. For our example we will consider the formula $(\neg p) \wedge (\neg q)$. In order to write the truth table for this formula, we need to first identify the subformulas, i.e. the smaller logical statements that make up the overall formula $(\neg p) \wedge (\neg q)$. Based on how the brackets are arranged, we see that there are only two smaller pieces:

- Piece 1: the subformula $\neg p$,

- Piece 2: the subformula $\neg q$.

These are then combined into the overall formula $(\neg p) \wedge (\neg q)$ using a conjunction. So, in order to figure out what this conjunction is doing, we need to first understand how $\neg p$ and $\neg q$ behave. The truth table will therefore look like:

p	q	$\neg p$	$\neg q$	$(\neg p) \wedge (\neg q)$
F	F			
F	T			
T	F			
T	T			

If we fill in the parts of the table corresponding to the negations *first*, then we can use our answers for that to figure out how to compute the $\wedge$ part.

Since negating a statement always flips truth-values, we can fill in the first two columns of our truth table quite easily. To complete the column for $\neg p$, we simply look at the column for p, inspect the T or F in each row, and then write the opposite in the column for $\neg p$. Reading from Row 1 downward, the column for p reads "F F T T". The column $\neg p$ will flip these values, so it will therefore read "T T F F".

The exact same reasoning will apply to $\neg q$. Again, we just look at the values listed in the column associated to q, and then write the opposite in the column for $\neg q$. Reading down the column for q, we see it reads "F T F T". Inverting all of these, the column for $\neg q$ will read "T F T F".

Our truth table is now partially completed:

p	q	$\neg p$	$\neg q$	$(\neg p) \wedge (\neg q)$
F	F	T	T	
F	T	T	F	
T	F	F	T	
T	T	F	F	

and we are left with evaluating the final column. The important point here is to recognise which connective is being used – in this case we are trying to figure out what the *conjunction* is doing in the formula $(\neg p) \wedge (\neg q)$. Crucially, we must now remember what the truth table for conjunction looks like.

Looking back at Section 2.3.1, we see that the conjunction of two statements is only true whenever both statements are themselves true. In other words, a conjunction is *false* if one of the two statements is false. Perhaps the most confusing part is that the truth table in Section 2.3.1 doesn't look exactly like what we have here. But, that is because we are now considering the conjunction of *two different things*: we don't care about $p \wedge q$ right now, we care about $(\neg p) \wedge (\neg q)$. In our truth table, these are the two columns that we should be looking at:

p	q	$\neg p$	$\neg q$	$(\neg p) \wedge (\neg q)$
F	F	T	T ⟶	?
F	T	T	F ⟶	?
T	F	F	T ⟶	?
T	T	F	F ⟶	?

For Row 1, we see that we have that $\neg p$ is true and $\neg q$ is also true. This is Row 4 of the truth table in Section 2.3.1, i.e. this is precisely the condition that is needed to make the conjunction $(\neg p) \wedge (\neg q)$ true. So, the first row of the final column will be a "T". The only way that the conjunction $(\neg p) \wedge (\neg q)$ is true is when *both* $(\neg p)$ and $(\neg q)$ are true. We see that in Rows 2,3 and 4 there is always at least one "F" in the columns for $(\neg p)$ and $(\neg q)$. So, the conjunction will be false in all 3 of these cases. Therefore, the column for the formula $(\neg p) \wedge (\neg q)$ will read "T F F F":

p	q	$\neg p$	$\neg q$	$(\neg p) \wedge (\neg q)$
F	F	T	T	T
F	T	T	F	F
T	F	F	T	F
T	T	F	F	F