Deduction
by Daniel Bonevac

Chapter 2 Sentences

Sentential Logic

- This chapter develops the basics of sentential logic. Sentential logic uses symbols to represent entire English sentences. So, a sentence like:

  - Clowns are scary.

  would be represented by a single symbol like ‘p’. This sentence is a simple or “atomic” sentence. Other sentences are compound or ‘molecular’, meaning that they are composed of two or more components. For example:

  - Clowns are scary <u>and</u> most kids are frightened by them.

- This is actually two sentences joined by the word ‘and’. We’ll see that sentential logic represents this compound sentence with two letters joined by a symbol that has the same function as the word ‘and’, for example: (p & q). The word ‘and’ is called a ‘connective’ which makes sense in this context because ‘and’ clearly connects the two simple sentences together.

Compound sentences and connectives.

- Interestingly, though, this is also a compound sentence:
  - Clowns are not funny.
- The reason this is a compound sentence is that it is a combination of the simple sentence “Clowns are funny” and the word ‘not’ which is also a connective.
- We call ‘not’ a ‘singulary connective’ because it forms a compound out of itself and a simple sentence. We call ‘and’ a ‘binary connective’ because it makes a compound out of itself and two sentences. We can speak more generally of ‘n-ary’ connectives, where n would be the number of simple sentences the connectives make into a compound sentence.
- At first it is a little weird to think of ‘not’ as a connective, because it really doesn’t seem to connect anything the way ‘and’ does. You’ll get used to it.

Truth-functional connectives

- There are lots of different connectives in English (see p. 37 for a short list of them) but we're only concerned with a subset of them that are truth-functional.
- To understand this term, you can think about mathematics again. Recall from high school algebra that an expression like this is sometimes called a function:
  - $y=x^2$
- Basically, the idea of a mathematical function is that if you plug a number in for x a specific number will come out for y. In the above function, if you plug 3 in for x, you get y = 9. If you plug 10 in for x you get y= 100, etc.
- x^2 is a singulary mathematical function, but there are n-ary mathematical functions as well. For example, mathematical addition is a binary function. In the expression x+y, the '+' sign connects the numbers very like the way 'and' connects sentences. The '+' sign is a binary function, because if you input any two numbers for x and y, e.g., 2 and 3, you get one particular number out, namely 5.
- The difference between a mathematical function and a truth function is this: mathematical functions take numbers as inputs and give you numbers as outputs. Truth functions take truth-values as inputs and give you truth-values as outputs.

Truth values and truth functions

- In classical logic there are only two truth values: true and false.
- A truth functional connective takes the truth value of it’s component sentences as inputs, and produces another (though not necessarily different) truth value as the output.
- We already noted that ‘not’ is a singulary truth-functional connective. It is now easy to see why this is the case.
- Reconsider the simple sentence:
  - Clowns are scary.
- This sentence can be either true or false. If it has the value ‘true’, then the sentence
  - Clowns are <u>not</u> scary.
- is false. On the other hand, if the original sentence is false, then the negated sentence will be true.
- So ‘not’ is a truth-function that converts a true sentence into a false one and a false sentence into a true one.

Truth functions and their symbols

- Here is the formal definition of a truth function (p. 40)
  - An *n-ary truth function* is a function taking *n* truth values as inputs and producing a truth value as an output.
- In symbolic logic we represent truth functions with truth tables. There are four basic truth functions that we will are concerned with in sentential logic. They correspond roughly to the English words : ‘not,’ ‘and,’ ‘or,’ and ‘if..then’.
- When we are speaking technically we do not use these words, however. Rather, we use the words ‘negation,’ ‘conjunction,’ ‘disjunction’ and ‘conditional’. These truth functions are represented with particular symbols as follows:

not	negation	¬
and	conjunction	&
or	disjunction	v
if..then	conditional	→

Truth tables: negation

- Truth functions are represented by truth tables. The truth table for negation, our only singulary truth-functional connective, is:

$\mathcal{A}$	$\neg\mathcal{A}$
T	F
F	T

- This table says simply that when you negate a true sentence you get a false one and vice versa.

Truth tables: conjunction

- The truth table for conjunction has more rows because conjunction is a binary connective. In general, an n-ary connective has 2^n rows in the truth table. So a binary connective like conjunction has 4 lines in its truth table.

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}$ & $\mathcal{B}$
T	T	T
T	F	F
F	T	F
F	F	F

- The truth table for conjunction represents the fact that a statement of the form (A & B) is true only when A is true and B is true, but false in all other circumstances.

Discussion of negation and conjunction

- The truth table for ‘¬’ reflects the way the word ‘not’ is ordinarily used in English. It does not reflect every usage, of course. For example, we sometimes say ‘no’ in an ironic way, to mean yes.
- The truth table for ‘&’ also reflects the usual way of using the English word ‘and’. There are other ways of using the word ‘and’ that it does not capture, however. For example:
  - Fred got in his car and drove home.
- This sentence actually means that Fred <u>first</u> got in his car and soon <u>afterwards</u> drove away in <u>his</u> car. We would not ordinarily say that the above sentence is true if Fred drove home in your car and then, on arriving, crawled into his own car and took a nap. Yet, if this is what happened the two simple sentences “Fred got in his car,” & “Fred drove home,” are true in this case as well, so the original sentence would be counted true in sentential logic.
- All this means is that the ‘&’ of sentential logic doesn’t mean everything that the ‘and’ of English can mean.

Truth table: disjunction

- This is the truth table for disjunction.

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}$ v $\mathcal{B}$
T	T	T
T	F	T
F	T	T
F	F	F

- It represents the fact that a statement of the form (A v B) is false only when A is false and B is false, but true in all other circumstances.

Discussion of disjunction

- The meaning of the disjunction 'v' is also slightly more restricted than the English word 'or'.
- Generally speaking we realize that when we say something of the form (A v B), only one of the simple sentences has to be true in order for the entire statement to be true. For example, suppose you were to tell Butch:
  - You will stop staring at me or I will poke you in the eye with my pencil.
- Butch would understand this statement to be true as long as one of these are true:
  - Butch stops staring at you.
  - You poke Butch in the eye with a pencil.
- What's a little weird is that the truth table for 'v' counts the original disjunction true if <u>both</u> A and B are true. In other words, the original disjunction is true even if Butch stops staring at you and you poke him in the eye anyway.
- You might think that this is a peculiar way to characterize disjunction, but actually it's not. It's true that in English we do sometimes use sentences of the form "A or B" to mean "<u>Either</u> A or B, but <u>not</u> both". This is what we call the <u>exclusive</u> use of the term. However, we also commonly use it in an <u>inclusive</u> sense as well. For example, the sentence:
  - Your reckless driving is going to get us hurt or arrested.
- Is clearly true even if your reckless driving gets us <u>both</u> hurt and arrested. So the truth table for disjunction captures the inclusive sense of 'or' not the exclusive sense.

Truth tables: conditional

- The truth table for the conditional is pretty fascinating. It looks like this.

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \rightarrow \mathcal{B}$
T	T	T
T	F	F
F	T	T
F	F	T

- In English, conditionals are statements of the form “If A, then B.” Conditionals have two parts, the ‘if A’ clause is called the ‘antecedent’. The ‘then B’ clause is called the ‘consequent’. The truth table says that the only time a conditional is false is when the antecedent is true and the consequent is false. It is true in all other circumstances.

Discussion of conditional (part 1)

- Consider a statement like:
  - If you keep staring at me, I will poke you in the eye with this pencil.
- You'll notice that if Butch keeps staring and yet you <u>don't</u> poke him in the eye with the pencil, then the original statement you made is false. So the second line of the truth table makes complete sense.
- We'd also be inclined to say that if Butch keeps staring at you and you <u>do</u> poke him in the eye with the pencil, then the statement you made is true. (A problem can be raised here, but we'll leave it to a later time.)
- Our difficulty in understanding the conditional stems from the last two lines of the truth table, when the antecedent is false. If, for example, Butch stops staring at you and you poke him in the eye anyway this might strike Butch as being a bit dishonest. Did you lie? Well, not really. You didn't say you <u>wouldn't</u> poke him in the eye if he stopped staring, only that you <u>would</u> poke him if he kept doing it.
- Even so, it seems a little weird to say that the statement is actually <u>true</u> in such a case. Most of us want to say that if all we know is that the antecedent is false, then we really don't know whether the whole conditional is true or false . Consider another example:
  - If you ask Renata out, Bruno is going to be sad.
- So suppose you don't ask Renata out. According to the truth table, this means the conditional is automatically true. That's very strange. Why should that be?

Truth functionality redux

- There are a few things we can say in defense of this assignment of truth values to the conditional. The first thing you have to understand is that sentential logic is truth-functional logic. This means that every connective we use has to be such that some assignment of truth values to the atomic sentence always results in some truth value for the compound sentence. In other words, it's not an option to leave some of the boxes in the table blank.
- Given this, the next question is whether it would make sense to fill out the table differently. For example, would it makes sense to say that the conditional is false whenever the antecedent is false? Would it makes sense, in other words, to fill out the table like this?

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \rightarrow \mathcal{B}$
T	T	T
T	F	F
F	T	F?
F	F	F?

Disjunctions and conditionals

- That answer is that it wouldn't makes sense, since it's just the truth table for '&.' '&' and '→' definitely don't mean the same thing.
- There is an interesting connection between → and v which can help us see the virtue of the original assignment of truth values, however. Recall this disjunction:
  - You will stop staring at me or I will poke you in the eye with my pencil.
- Notice that this is the same as saying:
  - If you don't stop staring at me, then I'm going to poke you in the eye with my pencil.
- This means that statements of these two forms should have the same truth tables.
  - $A \vee B$
  - $\neg A \rightarrow B$
- Interestingly, this is just what you get with the standard assignment of truth values to the conditional.

Comparison of truth table for v and →

- The equivalence of (A v B) and (¬A → B) can be shown with truth tables as follows. The first truth table is just the standard truth table for v. The second truth table is the standard truth table for → except that the values for ¬A are used instead of the values for A. (The values for ¬A must, of course, be the exact opposite of the values for A). The equivalence is demonstrated by the fact that the truth values in both yellow columns are the same.

$\mathcal{A}$	$\mathcal{B}$	$(\mathcal{A} \vee \mathcal{B})$
T	T	T
T	F	T
F	T	T
F	F	F

$\neg\mathcal{A}$	$\mathcal{B}$	$(\neg\mathcal{A} \rightarrow B)$
F	T	T
F	F	T
T	T	T
T	F	F

A fifth connective, the biconditional $\leftrightarrow$

- Now that we have learned the four basic connectives, we'll learn one more, known as the biconditional. The biconditional has the same truth value as the conjunction of two conditionals. In other words
  - (A $\leftrightarrow$B) means the same things as (A $\rightarrow$B) & (B $\rightarrow$ A)
- The truth table for the biconditional is

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \leftrightarrow \mathcal{B}$
T	T	T
T	F	F
F	T	F
F	F	T

Other connectives

- There are many other connectives in English that are simply not truth functional under any reasonable interpretation. For example, the word ‘because’ is a binary connective that is not remotely truth functional.
- On the other hand, some other English connectives can be captured in terms of the ones we have just defined.
- For example
  - “A but B” just means $(A \& B)$
  - “A unless B” just means $(\neg A \rightarrow B)$
  - “A only if B” just means $(A \rightarrow B)$
  - “A whenever B” just means $(B \rightarrow A)$

Advice for beginners

- You may find it very confusing to think a lot about the relation between the truth functional connectives we've introduced here and the ordinary English words they are correlated with.
- The good news is that you don't really have to think about this if you don't want to. But you really do need to learn what the truth functional connectives mean, and this means memorizing their truth tables.

How symbolic languages are created

- We're just about ready to define a symbolic language of sentential logic. But first we need to be clear about what's involved in this task.
- Competent use of any anguage depends on understanding two different things about it.
- First, we need to understand how sentences of the language are constructed. This is called the <u>syntax</u> of the language, and it requires us to define the <u>vocabulary</u> and the <u>formation rules</u> of the language.
- Second, we need to understand what the words and sentences actually mean. This is called the <u>semantics</u> of the language. We have actually already provided the semantics of our sentential language by defining the truth tables for the connectives. The truth tables tell us under what conditions sentences employing these connectives are true, and in logic that's actually what it means to say what a sentence means.

Vocabulary

- The vocabulary of our sentential language may be defined as follows: (p. 47)
  - Sentence letters: p, q, r, s, p_1, q_1, r_1,s_1, p_1, r_1, ..etc.
  - Connectives: ¬, v, &, →, ↔
  - Grouping indicators: (,)
- The formation rules are specified as follows:
  - Any sentence letter is a formula.
  - If $\mathcal{A}$ is a formula, then $\neg\mathcal{A}$ is a formula.
  - If $\mathcal{A}$ and $\mathcal{B}$ are formulas, then ($\mathcal{A}$ & $\mathcal{B}$), ($\mathcal{A}$ v $\mathcal{B}$), ($\mathcal{A}\rightarrow\mathcal{B}$) and ($\mathcal{A}\leftrightarrow\mathcal{B}$) is a formula.
  - Every formula can be constructed by a finite number of applications of these rules.

Object language vs. Metalanguage.

- You may have noticed that in stating the formation rules we used capital letters in a different font. For example, the formation rule for negation said:
  - If $\mathcal{A}$ is a formula, then $\neg\mathcal{A}$ is a formula.
- This isn't just for show. The rules are stated in what we call a <u>metalanguage</u>. The metalanguage is the language we used to talk about the language being defined, which is called the <u>object language</u>.
- So, when we write something like ($\mathcal{A}$ & $\mathcal{B}$), this is not an actual formula of the object language. On the other hand all of these sentences are formulas of the object language that we say have the <u>form</u> ($\mathcal{A}$ & $\mathcal{B}$).
  - (p & q)
  - (p_1 & p_2)
  - (p & ($p_1 \to p_2$))
  - (p v q) & (r $\to$ t)

Scope and main connectives

- The scope of a connective is defined as the connective itself plus the formulas that it links together.
- The main connective is the formula with the largest scope. In other words, it is the connective that holds the whole formula together.
- Let's look at some quick examples.
- What is the scope of each of the connectives in the following formula?
  - $((\neg p \ \& \ (q \lor r)) \leftrightarrow \neg(s \ \& \ p_1))$
- What is the main connective in the following formulas?
  - $\neg(p \ \& \ q)$
  - $(\neg p \rightarrow q)$
  - $(\neg p \ \& \ (q \lor r))$
  - $\neg p \leftrightarrow \neg p$
  - $\neg\neg\neg p$

Scope and main connectives

- The answers are as follows:
- What is the scope of each of the connectives in the following formula?
  - $((\neg p \ \& \ (q \lor r)) \leftrightarrow \neg(s \ \& \ p_1))$
    - Scope of first $\neg$ = $\neg p$
    - Scope of first & = $(\neg p \ \& \ (q \lor r))$
    - Scope of v = $(q \lor r)$
    - Scope of $\leftrightarrow$ = entire formula
    - Scope of second $\neg$ = $\neg(s \ \& \ p_1)$
    - Scope of second & = $(s \ \& \ p_1)$
- What is the main connective in the following formulas?
    - $\neg(p \ \& \ q) = \neg$
    - $(\neg p \rightarrow q) = \rightarrow$
    - $(\neg p \ \& \ (q \lor r)) = \&$
    - $\neg p \leftrightarrow \neg p = \leftrightarrow$
    - $\neg\neg\neg p$ = first $\neg$

Recognizing formulas

- Here is some quick practice to determine whether we know the formation rules of the language. Are the following in accord with the formation rules for our sentential language?
- $\mathcal{A}$ & $\mathcal{B}$
- *No, this is in the metalanguage.*
- p v q
- No, this is missing the outside parenthesis. It should be (p v q)
- &p
- No, & is a binary connective.
- ¬(¬p)
- No, parentheses aren't used to negate a negation.
- ¬(¬p & q)
- Yes. The outside negation is the main connective. It is negating the conjunction.
- (p & q) → (r v t).
- No, it is missing the outside parenthesis and 't' is not an official formula.

Some unofficial conventions

- In practice we liberalize the official conventions. Specifically:
  - It’s ok to use other sentence letters as formulas, but stay away from the back of the alphabet, x-z.
  - It’s ok to use square brackets, [,] and braces {,} when lots of parentheses are needed.
  - You can drop the outermost brackets.

Truth tables, interpretations and decision procedures

- Here are a couple of other handy terms:
  - An <u>interpretation</u> of a formula is an assignment of truth values to its sentence letters.
  - A <u>decision procedure</u> for some property P is a mechanical, infallible method for determining whether something has P.
- Because sentential logic is truth functional, a truth table gives us a decision procedure for determining whether an interpreted sentence is true or false.

Example of truth table as decision procedure

- Consider the following interpreted sentence. What is it’s truth value?
  - Formula (p v q) $\rightarrow$ (p &q)
  - p=T, q=F,
- You can set it up in a table like this. So, first assign T to all the p’s and F to all the Q’s as required by the interpretation above.

(p	v	q)	$\rightarrow$	(p	&	q)
T		F		T		F

Example of truth table as decision procedure

- Now you need to notice that the $\rightarrow$ is the main connective in this formula. So the truth value of the sentence will be a function of the two formulas the $\rightarrow$ is joining together, namely (p v q) and (p & q).
- So first compute the truth values of these two formulas. Notice you will <u>not be able to do this</u>, unless you know the truth tables for v and &. If you do know these truth tables, then you will see that the correct assignments are:

(p	v	q)	$\rightarrow$	(p	&	q)
T	T	F		T	F	F

Example of truth table as decision procedure

- Now all you have to do is compute the truth value of the $\rightarrow$. This is a function of the truth values you have just assigned to (p v q) and (p & q).
- Again, you must have memorized the truth table for the $\rightarrow$ to be able to do this easily. If you have memorized the truth table, then you know that if $\mathcal{A}$ is T and $\mathcal{B}$ is F, then the truth value of ($\mathcal{A} \rightarrow \mathcal{B}$) is F.
- Remember that when we use the upper case letters like $\mathcal{A}$ and $\mathcal{B}$, we are using the meta-language. These letters can stand for <u>any formula at all</u>. Here, $\mathcal{A}$ stands for (p v q) and $\mathcal{B}$ stands for (p & q).

(p	v	q)	$\rightarrow$	(p	&	q)
T	T	F	F	T	F	F

Example of truth table as decision procedure 2

- Consider the following interpreted sentence. What is it's truth value?
  - Formula ¬((p & q) → (r v q))
  - p=f, q=f, r=t
- You can set it up in a table like this. Notice that in this case the main connective is ¬. So the truth value of the entire sentences will be the truth value under the ¬.

¬	((p	&	q)	→	(r	v	q))
	F		F		T		F

Example of truth table as decision procedure 2

- The next step of the evaluation will look like this, following from the truth tables for & and v.

¬	((p	&	q)	→	(r	v	q))
	F	F	F		T	T	F

Example of truth table as decision procedure 2

- The next step of the evaluation will look like this, following from the truth table for →.

¬	((p	&	q)	→	(r	v	q))
	F	F	F	**T**	T	T	F

Example of truth table as decision procedure 2

- The last step of the evaluation will look like this, following from the truth tables for ¬.

¬	((p	&	q)	→	(r	v	q))
F	F	F	F	T	T	T	F

Complete truth tables for sentences.

- Complete truth tables for sentences work just like complete truth tables for the sentence connectives. (p. 66-67) For example :
- $p \rightarrow (r \rightarrow p)$

p	r		p	$\rightarrow$	(r	$\rightarrow$	p)
T	T		T	T	T	T	T
T	F		T	T	F	T	T
F	T		F	T	T	F	F
F	F		F	T	F	T	F

What you can learn from truth tables

- A truth table gives you every possible interpretation of a sentence.
- A contradiction is a sentence that is false under every possible interpretation. So if every space in the column directly under the main connective has an F in it, the sentence is a contradiction.
- A valid or tautologous sentence is true under every possible interpretation. So if every space in the column directly under the main connective has a T in it, the sentence is a tautology.
- The sentence is contingent if there is at least one T and one F in the column directly under the main connective.
- The sentence is satisfiable if there is at least one T in the column directly under the main connective.

Deciding validity with truth tables

- You can determine whether an argument form is <u>deductively valid</u> by checking to see whether the truth table for the corresponding conditional is a tautology.
- For example, the following argument form is deductively valid.
  1. (p v q)
  2. <u>¬p   .</u>
  3. q
- You can show this by doing the truth table for the following conditional.
  - ((pvq) & ¬p) → q
- Notice that this sentence is simply the argument put in the form of a conditional. The premises are put together with the &, and the conditional → stands for the implication relation. If you do the truth table correctly, you'll get all T's in the column under the →. If there is even one F, then that means it's possible for the premises to be true and the conclusion false. In other words, the argument is invalid.

Deciding logical equivalence

- Truth tables can also be used to determine whether two formulas are logically equivalent. Formulas are logically equivalent if they have exactly the same truth conditions. So, one way to check for logical equivalence is to do two different truth tables and check to see that the columns under the main connective are identical.
- Another, slightly easier way, is to connect the two formulas with a ↔ and run a truth table on the resulting formulas. If the column under the ↔ has all T's, then the biconditional is a tautology, which means that the two formulas are logically equivalent.

Example of logical equivalence test

- Are these two formulas logically equivalent?
  - ¬(p v q)
  - (¬p & ¬q)
- Check the truth table for this biconditional
- ¬(p v q) ↔ (¬p & ¬q)

p	q		¬	(p	v	q)	↔	(¬p	&	¬q)
T	T		F	T	T	T	T	F	F	F
T	F		F	T	T	F	T	F	F	T
F	T		F	F	T	T	T	T	F	F
F	F		T	F	F	F	T	T	T	T

Truth tables with 3 sentence letters.

- Truth tables with n sentence letters have 2^n rows. That means that a truth table for a formula with 5 sentence letters will have 32 rows. This makes using truth tables impractical, which is why we need to develop other proof methods for sentential logic. But it is a right of passage to do at least one truth table with more than four rows. So let's do a truth table for the following formula:

- [(p ↔ q) & (q ↔ r)] & (p & ¬r)

p	q	r	[(p	↔	q)	&	(q	↔	r)]	&	(p	&	¬r)
T	T	T	T	T	T	T	T	T	T	F	T	F	F
T	T	F	T	T	T	F	T	F	F	F	T	T	T
T	F	T	T	F	F	F	F	F	T	F	T	F	F
T	F	F	T	F	F	F	F	T	F	F	T	T	T
F	T	T	F	F	T	F	T	T	T	F	F	F	F
F	T	F	F	F	T	F	T	F	F	F	F	F	T
F	F	T	F	T	F	F	F	F	T	F	F	F	F
F	F	F	F	T	F	T	F	T	F	F	F	F	T

Interpreting English sentences.

- Translations of English arguments into sentential logic are of more interest when we start doing proofs, but it's worth doing a few here just to get the hang of it.