

Propositional logic

Michael Franke

Syntax & semantics of propositional logic; truth-tables; tautologies vs. contradictions vs. contingencies; logical equivalence; translations from natural language into propositional logic; semantic meaning vs. pragmatic enrichment; argument schemas & logical validity.

1 The language of propositional logic

Propositional logic (PROPLOG) studies how propositions are combined by logical operators, which closely correspond to certain sentential connectives in natural language (such as *and*, *or*, *if*, or *not*). A *proposition* in the sense of PROPLOG is a minimal unit of thought which can be evaluated as true or false independently of other propositions.¹ For example, the logical structure of the sentence:

$\underbrace{\text{The earth is round}}_p \text{ and } \underbrace{\text{the moon is made of cheese.}}_q$

could be analyzed as composed of two propositions, viz., the proposition (denoted here with *proposition letter* p) that the earth is round and the proposition (denoted by q) that the moon is made of cheese. These two propositions are connected by a logical operator “and,” for which we write \wedge in PROPLOG. The logical structure of the complex sentence above can therefore be written as $p \wedge q$ in PROPLOG.

¹ The notion of a proposition in this sense is not unproblematic. For example, a case like “*This pixel is red.*” seems like a minimal unit of truth-evaluable information about the color of a particular pixel, but it is not independent of another statement like “*This pixel is blue.*” (Historically, this color-related problem was brought up famously by Frank Ramsey in response to the early logical work of Ludwig Wittgenstein.)

1.1 Proposition letters & sentential connectives

The language of PROPLOG is formed by:

- (i) a set of *proposition letters* $\mathfrak{P} = \{p, q, r, s, p_1, q_{27}, \dots\}$, and
- (ii) a set of *sentential connectives* $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$.²

The sentential connectives have names and are intended to correspond (approximately) to natural language paraphrases:

name	paraphrase	symbol
negation	“not”	\neg
conjunction	“and”	\wedge
disjunction	“or”	\vee
implication	“if . . . , then . . . ”	\rightarrow
equivalence	“if and only of”	\leftrightarrow

² You will also find terminology like *logical connectives* or *logical operators*. There may be additional connectives used by some logicians or textbooks, and you might find slightly different symbols for the same notions in some places

1.2 Formulas

The language \mathcal{L} of PROPLOG is the set of all *formulas* which are recursively defined as follows:³

- (i) Every proposition letter is a formula.
- (ii) If φ is a formula, so is $\neg\varphi$.
- (iii) If φ and ψ are formulas, so are:
 - a. $(\varphi \wedge \psi)$
 - b. $(\varphi \vee \psi)$
 - c. $(\varphi \rightarrow \psi)$
 - d. $(\varphi \leftrightarrow \psi)$
- (iv) Anything that cannot be constructed by (i)–(iii) is not a formula.

Examples for formulas of PROPLOG are:⁴

$$\begin{array}{ll} p & p \wedge p \\ p \rightarrow \neg q & (p \vee q) \leftrightarrow r \end{array}$$

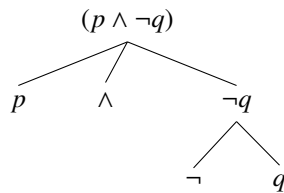
Examples of strings made of proposition letters and logical connectives which are *not* formulas of PROPLOG are:

$$\begin{array}{ll} (p) & \neg pq \\ p\neg \rightarrow \neg q & p \vee q \leftrightarrow r \end{array}$$

1.3 Syntactic trees

The recursive definition for formulas of PROPLOG gives an internal structure to each formula. Take the example $p \wedge \neg q$. There is only one way in which this formula could have been generated by a constructive process that follows the recursive definition above. In the last step of that process, the two subformulas $\varphi = p$ and $\psi = \neg q$ have been combined to form an expression of the form $\varphi \wedge \psi$ using rule (iii) part a. The first subformula $\varphi = p$ is constructed by rule (i). The second formula $\psi = \neg q$ can only be constructed by first using rule (i) to introduce q and then using rule (ii) to introduce the negation sign.

A *syntactic tree* is a useful visual illustration of the internal structure of a formula. The syntactic tree of formula $p \wedge \neg q$ is this:



The construction of a complex formula, and therefore its syntactic tree, is always recoverable by following the introduction of the parentheses in step (iii). This is illustrated by the minimal pair in Figure 1.⁵

³We will make consistent use of Greek letters $\varphi, \psi, \chi, \dots$ as variables for formulas.

⁴We conventionally omit the outermost parentheses of a formula.

⁵This is why at most the outer parentheses may be omitted, but never any other pair of parentheses.

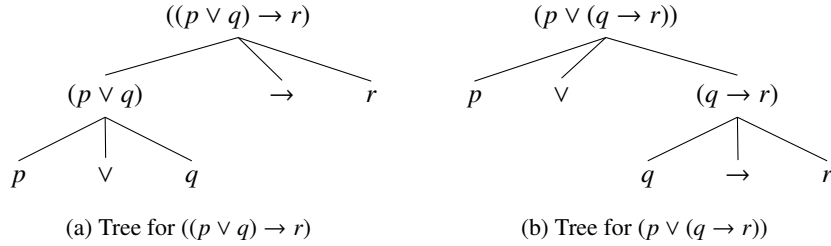


Figure 1: Examples of syntactic trees

1.4 Terminology

A formula of **PropLog** which consist of a single proposition letter is called *atomic formula*. Any formula of **PropLog** which is not atomic is called a *complex formula*.

Each complex formula has a *main connective*. The main connective is the last sentential connector introduced during the construction of the formula. A complex formula is also often called by the name of its main connective. For example, the formula $p \wedge \neg q$ has a conjunction as its main operator and would therefore be called a conjunction. The formula $(p \vee q) \rightarrow r$ from Figure 1(a) is an implication, while the formula $p \vee (q \rightarrow r)$ from Figure 1(b) is a disjunction.

For some connectives, the subformulas that they contain may have special names as well. In the conjunction $\varphi \wedge \psi$, the subformulas φ and ψ are called *conjuncts*. In the disjunction $\varphi \vee \psi$, the subformulas φ and ψ are called *disjuncts*. In the implication $\varphi \rightarrow \psi$, the subformula φ is called *antecedent* and the subformula ψ is called *consequent*.

Exercise 1. Determine which of the following strings are formulas of propositional logic. For any formula, determine its main operator.

- | | |
|--------------------|--|
| a. q_{12} | d. $p \rightarrow (p \wedge p)$ |
| b. $p, q \wedge r$ | e. $(p \rightarrow)(p \wedge p)$ |
| c. $(p) \wedge q$ | f. $(p \vee \neg q) \leftrightarrow (r \rightarrow (\neg(p \vee \neg p)))$ |

Exercise 2. Draw the syntactic tree for each of the following formulas.

- | | |
|--------------------------|--|
| a. $p \leftrightarrow q$ | c. $p \rightarrow \neg(q \wedge r)$ |
| b. $\neg p \wedge p$ | d. $(\neg p \vee \neg q) \wedge (r \rightarrow p)$ |

2 The semantics of propositional logic

The language of **PROPLOG** defines rules of systematically combining proposition letters with sentential connectives. The *semantics* of **PROPLOG** gives a meaning to each formula. Yet, **PROPLOG** is not concerned with the meaning of the proposition letters. Whether the proposition p corresponding to “The earth is round.” is true or false in this world we live in, is of no concern to the logician. The (propositional) logician cares only about the meaning of the sentential connectives. More specifically, **PROPLOG** pays attention to a specific kind of meaning, namely *truth-conditional meaning*. Concretely, **PROPLOG** analyzes the meaning of a sentential connective $\odot \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ in terms of how the truth or falsity of the complex sentence $\varphi \odot \psi$ depends on the truth or falsity of its components φ and ψ ; and how the meaning of $\neg\varphi$ depends on the truth or falsity of φ .

2.1 Truth tables

One way of defining the semantics of the sentential connectives is by *truth tables*. Truth tables are also handy for systematically calculating the conditions under which complex formulas are true.

Take the complex formula $\neg\varphi$.⁶ For all we know, φ could be either true or false. There are just these two possibilities.⁷ To give a truth-conditional meaning to the operator \neg we need to define whether $\neg\varphi$ is true or false for each case: when φ is true, and when φ is false. The truth table that defines the meaning of negation is given in Table 1(a).⁸ **PROPLOG** treats negation like a switch: take a formula φ as input, look at its truth value, and swap it for the other (since there are only two truth values).

Other sentential connectives than negation take two formulas as input, so to speak, and may therefore be conceived of as functions that take a pair of truth values as input to return a single truth-value. The semantics of the binary connectives is given in Table 1(b):

φ	$\neg\varphi$	ϕ	ψ	$\phi \wedge \psi$	$\phi \vee \psi$	$\phi \rightarrow \psi$	$\phi \leftrightarrow \psi$
1	0	1	1	1	1	1	1
1	0	1	0	0	1	0	0
0	1	0	1	0	1	1	0
0	1	0	0	0	0	1	1

(a) Semantics of negation

(b) Semantics of binary connectives

⁶We use φ here as a variable which could represent any formula, no matter if it is a proposition letter or extremely long and complex.

⁷That **PROPLOG** only considers two truth values is also referred to as the *principle of bivalence*. There are other logics, so-called many-valued logics, which allow for more than two truth values, e.g., to represent degrees of truth or uncertainty.

⁸We here write 1 for the truth value “true” and 0 for “false.” It is also common to use letters T and F , or yet other symbols in truth tables.

Table 1: Semantics of sentential connectives in propositional logic expressed in terms of truth tables

2.2 Working with truth tables

Truth tables are useful for systematically working out the conditions under which complex formulas are true. Consider the complex formula $(\varphi \wedge \psi) \rightarrow$

$\neg\chi$. We do not know whether the subformula φ , ψ and χ are atomic or themselves complex. But we want to know under which truth value assignments to φ , ψ and χ the formula $(\varphi \wedge \psi) \rightarrow \neg\chi$ is true or false. In order to find out, we can construct a truth table that starts by enumerating all logical possibilities of truth-value assignments for the formulas φ , ψ and χ (the first three columns in the table below). We then work towards the “goal formula” $(\varphi \wedge \psi) \rightarrow \neg\chi$ by following the recursive definition of formulas of **PRO-LOG**. At each step we apply the definition of the semantics of the relevant sentential connective until we arrive at the “goal formula”. Following this method, we construct the truth table in Table 2, which shows that the formula $(\varphi \wedge \psi) \rightarrow \neg\chi$ is always true, except when all three subformulas, φ , ψ and χ are all true at the same time.

φ	ψ	χ	$\varphi \wedge \psi$	$\neg\chi$	$(\varphi \wedge \psi) \rightarrow \neg\chi$
1	1	1	1	0	0
1	1	0	1	1	1
1	0	1	0	0	1
1	0	0	0	1	1
0	1	1	0	0	1
0	1	0	0	1	1
0	0	1	0	0	1
0	0	0	0	1	1

Table 2: Truth table for formula $(\varphi \wedge \psi) \rightarrow \neg\chi$

2.3 Contingencies, tautologies & contradictions

A common application of truth tables is to find out whether a given formula is always true, always false or whether it can sometimes be true and sometimes be false. The truth table in Table 2 shows that the formula $(\varphi \wedge \psi) \rightarrow \neg\chi$ belongs to the last category: there are cases, i.e., assignments of truth values to its components (= rows in the table), where it is true, and there are cases where it is false. A formula which can become true and false for different assignments of truth values to its components is called a *contingency*. A formula which is always true under any constellation of truth values for its components is called a *tautology*. A formula which is false for all ways of assigning truth values to its subformulas is a *contradiction*. We say that a tautology is a formula that is *necessarily true*, or *true by logical necessity*. Similarly, a contradiction can be said to be *necessarily false*, or *false by logical necessity*.⁹

To show that any formula of the form $\varphi \vee \neg\varphi$ is a tautology, we can construct the relevant truth table, shown in Table 3(a), and check whether the column for the “goal formula” contains only ones, no zeros. Similarly, the truth table in Table 3(b) reveals that any formula of the logical form $\varphi \wedge \neg\varphi$ is a contradiction.

⁹Another way of thinking about this is to imagine that you know nothing at all about some foreign universe. You don’t know at all which proposition letters are true or not. Given a tautology you can nevertheless be sure that it is true, no matter what the world is like. Of a contradiction you are sure that it is false. And for a contingency you know that you don’t know whether it is true or false, before you learn facts about the world.

φ	$\neg\varphi$	$\varphi \vee \neg\varphi$
1	0	1
0	1	1

 (a) Truth table for $\varphi \vee \neg\varphi$

φ	$\neg\varphi$	$\varphi \wedge \neg\varphi$
1	0	0
0	1	0

 (b) Truth table for $\varphi \wedge \neg\varphi$

Table 3: Truth tables for a tautology and a contradiction

2.4 Logical equivalence

Two formulas φ and ψ are logically equivalent if they have exactly the same truth value no matter how we assign meanings to all subformulas of φ and ψ , as long as we assign any subformula that occurs in both φ and ψ the same truth value. We can use truth table to test when or demonstrate that two formulas are logically equivalent. To do this, we just have to make a combined truth table in which we have one column for φ and another for ψ . φ and ψ are logically equivalent exactly when the truth values in the columns that correspond to them are identical in each row. The truth table in Table 4 shows that $\varphi \wedge \neg\psi$ and $\neg(\varphi \rightarrow \psi)$ are in fact logically equivalent.

φ	ψ	$\neg\psi$	$\varphi \wedge \neg\psi$	$\varphi \rightarrow \psi$	$\neg(\varphi \rightarrow \psi)$
1	1	0	0	1	0
1	0	1	1	0	1
0	1	0	0	1	0
0	0	1	0	1	0

 Table 4: Truth table showing the equivalence of $\varphi \wedge \neg\psi$ and $\neg(\varphi \rightarrow \psi)$

If two formulas φ and ψ are *not* logically equivalent, we can also show this with a truth table. In that case, there must be at least one row in which their truth values differ. In practice it may help to mark one such row as the refuting counterexample against the assumption of logical equivalence. This is shown in Table 5.

φ	ψ	$\varphi \rightarrow \psi$	$\varphi \leftrightarrow \psi$	counterexample
1	1	1	1	\Leftarrow
1	0	0	0	
0	1	1	0	
0	0	1	1	

 Table 5: Truth table showing the non-equivalence of $\varphi \rightarrow \psi$ and $\varphi \leftrightarrow \psi$

Exercise 3. For each of the following formulas, try to intuit whether it is a tautology, a contradiction or a contingency. Write down your best guess. Then use the truth-table method to find out for certain.

- | | |
|--|---|
| a. $\varphi \rightarrow \neg\varphi$ | c. $(\varphi \wedge \neg\varphi) \rightarrow \psi$ |
| b. $\varphi \leftrightarrow (\varphi \wedge \psi)$ | d. $(\varphi \wedge \neg\psi) \leftrightarrow (\varphi \rightarrow \psi)$ |

Exercise 4. For each of the following pairs of formulas, try to intuit whether they are logically equivalent or not. Write down your guess. Then use truth tables to determine whether they are. Highlight the relevant columns in your truth table. Mark a row as a counterexample if they are not.

- | | |
|--|---|
| a. $\varphi, \neg\neg\varphi$ | c. $\neg\varphi \vee \psi, \varphi \rightarrow \psi$ |
| b. $\varphi \leftrightarrow \phi, \varphi \wedge \psi$ | d. $\varphi \vee (\chi \rightarrow \neg\varphi), \psi \vee (\chi \rightarrow \neg\psi)$ |

3 Logical structure of meaning

Linguistic theory distinguishes between *semantic meaning* of a sentence or an expression and additional *pragmatic enrichments* to the literal meaning. The literal meaning is the context-independent logical core meaning. Pragmatic enrichments arise in context by taking into account general world knowledge or the reasons why a speaker used language in the way that they did. Pragmatic enrichments may be part of what we “perceive” to be the most likely interpretation of a sentence; they may be what the speaker meant to say, but not what the sentence means literally.¹⁰

Our current goal is to learn to excavate the logical structure of natural language sentences. Therefore, we have to learn to “strip off the pragmatic layer,” so to speak. Indeed, the logical operators of PROPLOG bear a close correspondence to natural language expressions *and*, *or*, *if* and *if and only if*. But the correspondence is not a perfect match. This is partly due to the difference between semantic meaning and pragmatic enrichment.¹¹ Let us look at a few interesting cases.

¹⁰The most obvious example is an utterance of a sentence like *The weather is so nice today!* when clearly meant ironically.

¹¹There are also arguments suggesting that the logical operators of PROPLOG are *not* good enough for analyzing the semantic meaning of their corresponding natural language expressions. Most importantly, this concerns the semantic meaning of conditional sentences, i.e., natural language sentences with *if dots, then ...*.

3.1 Conjunction & order-sensitivity

Logical conjunction is order-insensitive: $\varphi \wedge \psi$ and $\psi \wedge \varphi$ are logically equivalent. But natural language conjunctions are not necessarily. Saying that

They got married and had children.

might be understood differently from saying that

They had children and got married.

Still, we would analyze both of these sentences as having a logical-conjunctive meaning, additional inferences about the temporal order of events notwithstanding.

3.2 Disjunction: inclusive vs. exclusive

Logical disjunction is *inclusive*: $\varphi \vee \psi$ is true when φ and ψ are both true. Yet natural language disjunctions *can* be understood as exclusive disjunction according to which exactly one, but not both disjuncts are true. For example, if I say:

She owns a Porsche or a Ferrari.

you might take that to mean that she doesn’t own both. But did I *literally* say that? — The logical meaning of that sentence is more plausibly just analyzed as inclusive disjunction. This is because it is not contradictory to say:

She owns a Porsche or a Ferrari and possibly both.

This contrasts with the weirdness of saying:

She owns either a Porsche or a Ferrari and possibly both.

suggesting that natural language *either ... or ...* does have an exclusive disjunctive meaning, but simple *or* does not.

3.3 Conditional perfection

PROPLOG analyzes implication as so-called *material implication*, but it is controversial whether natural language conditional sentences might not have a much richer meaning, e.g., requiring a discernible causal-inferential relation between antecedent and consequent. Moreover, similar to the case of inclusive vs. exclusive disjunctions, many natural language conditionals receive a so-called *conditional perfection reading*, i.e., they are read like an “if and only if” statement which would correspond to logical operator \leftrightarrow , even though their logical meaning might just be that of a simple implication \rightarrow . For example, the conditional statement:

If Bubu comes, Kiki comes.

does suggest to a certain degree that Kiki *only* comes when Bubu does. But that is not the logical core meaning of this sentence, but rather a pragmatic enrichment on top of the logical meaning of the sentence. To see this, compare the difference in weirdness between saying

If Bubu comes, Kiki comes, and maybe Kiki comes no matter what.

which seems fine, especially when compared to the much weirder:

Kiki comes only if Bubu comes, and maybe Kiki comes no matter what.

3.4 Excavating the logical structure of natural language sentences

Despite superficial discrepancies, it is possible and highly informative to try to lay bare the logical structure of natural language sentences. The logical relation between propositions contained in a natural language sentence may sometimes be opaque. Take the following example sentence:

If drawing a red card means that I lost, I lost.

This sentence does have a propositional-logical structure, even though it is not at all apparent from the way it is realized in natural language. There are two atomic (truth-evaluable) propositions involved here:

p : I drew a red card.

q : I lost.

With this *translation key*, we can then determine the logical structure of the sentence above as:

$$(p \rightarrow q) \rightarrow q$$

Here are a few other examples.

- (i) John is hungry.
Key: p : John is hungry. Logical form: p
- (ii) Mary likes John.
Key: p : Mary likes John. Logical form: p
- (iii) John is hungry and Mary likes John.
Key: p : John is hungry. q : Mary likes John.
Logical form: $p \wedge q$
- (iv) Berlin is east of Hamburg and Bremen.
Key: p : Berlin is east of Hamburg. q : Berlin is east of Bremen.
Logical form: $p \wedge q$
- (v) Call me 'sweetie' once more and, no kidding, I'll step on your toe.
Key: p : I'm not kidding. q : I'll step on your toe.
 r : You call me 'sweetie' once more.
Logical form: $p \wedge (r \rightarrow q)$
- (vi) I will only play salsa, if you give me earplugs.¹²
Key: p : I will play salsa. q : You give me earplugs.
Logical form: $p \rightarrow q$

¹²There is a likely inclination to analyze this as $p \leftrightarrow q$, but it is not clear whether this is part of the semantic/logical content here. Suppose you know that the sentence is true. Now you learn that earplugs have been handed over. Does that *logically* entail that salsa was played? No, because the sentence only says that earplugs are a *necessary* but not necessarily *sufficient* condition for salsa.

Exercise 5. For each of the following sentences give the translation key and the (propositional) logical form.

- (a.) If John is hungry, then Mary likes John.
- (b.) If John is hungry and Bill runs, then Mary doesn't like John.
- (c.) John is hungry or it's false that Bill runs.
- (d.) Mary likes John or Mary doesn't like John.
- (e.) Mary likes John but John is hungry.
- (f.) Mary likes John provided Bill runs.

4 Valuation functions, possible worlds & logical entailment

4.1 Valuation functions

The truth-tables method introduced above is very practical, but a proper definition of the semantics of **PROPLOG** should better not rely on vocabulary like “rows” and “columns,” but use mathematical notions like “function” or “set.” The proper definition of the semantics of **PROPLOG** is therefore formulated in terms of so-called valuation functions.

A *valuation function* V is a function that assigns to every formula of propositional logic a unique truth value (0 or 1). More specifically, think of a valuation function as a way of assigning truth-values to all proposition letters. Once we know the truth or falsity of all proposition letters, the truth or falsity of all complex formulas is non-arbitrary, because it is fixed by the meaning of the sentential connectives.

$V(p), V(q), \dots$	arbitrary
$V(\neg\phi) = 1$	iff $V(\phi) = 0$
$V(\phi \wedge \psi) = 1$	iff $V(\phi) = 1$ and $V(\psi) = 1$
$V(\phi \vee \psi) = 1$	iff $V(\phi) = 1$ or $V(\psi) = 1$
$V(\phi \rightarrow \psi) = 0$	iff $V(\phi) = 1$ and $V(\psi) = 0$
$V(\phi \leftrightarrow \psi) = 1$	iff $V(\phi) = V(\psi)$

A valuation function is analogous to a row in a truth table, when we use proposition letters, not variables for formulas like φ or ψ .¹³

4.2 Possible worlds & truth-conditional meaning

A valuation function can also be thought of as a *possible world*.¹⁴ A possible world is a way in which things could be. A possible world conforms to the rules of logic, but otherwise allows all atomic propositions to be different from the facts in the *actual world*.

Using valuation functions or, equivalently, possible worlds, we can give a clear definition of the meaning of a formula (and, therefore by extension, to the logical meaning of a natural language sentence). The truth-conditional meaning of formula φ is the set of all valuation functions, or possible worlds, for which φ is true. Formal semantics often uses notation $\llbracket \varphi \rrbracket$ to denote the meaning of a formula in this sense:

$$\llbracket \varphi \rrbracket = \{V \mid V \text{ is a valuation function such that } V(\varphi) = 1\}$$

Alternatively, we could say that the meaning of a formula is the set of possible worlds in which it is true.¹⁵

4.3 Tautologies, contradictions, contingencies & equivalence (revisited)

Using valuation functions and/or possible worlds, we can also give a proper definition of notions we used previously in the context of truth tables. We

Table 6: Semantics of propositional logic in terms of valuation functions

¹³If we use truth tables for complex formulas built with Greek letters like φ or ψ , we cannot say that a row corresponds to a valuation function because we do not know what the variables like φ or ψ represent. For example, φ might be a tautology or a contradiction. We would still write out all possible truth-value assignments for φ , but we just cannot be sure if there is a valuation function that would correspond to each row. For this reason, when we look at argument schemas below, using truth tables to check for logical validity, we use proposition letters instead of variables for formulas.

¹⁴This terminology is widely used in formal semantics and philosophy.

¹⁵Compare the famous passage from Wittgenstein’s *Tractatus Logica Philosophicus*: “Einen Satz verstehen, heißt, wissen was der Fall ist, wenn er wahr ist. (Man kann ihn also verstehen, ohne zu wissen, ob er wahr ist.)” [TLP 4.024]

say that φ is a tautology iff $V(\varphi) = 1$ for all valuation functions V . We say that φ is a contradiction iff $V(\varphi) = 0$ for all valuation functions V . We say that φ is a contingency iff there is at least one valuation function V_1 such that $V_1(\varphi) = 1$ and there is at least one valuation function V_2 such that $V_2(\varphi) = 0$. We say that φ and ψ are logically equivalent, iff for all valuation functions $V(\varphi) = V(\psi)$.

4.4 Logical entailment & informativity

Similarly, we can now give a proper definition of logical entailment. Formula φ logically entails ψ iff for all valuation functions V , if $V(\varphi) = 1$, then $V(\psi) = 1$. An alternative formulation is this: φ logically entails ψ iff $\llbracket \varphi \rrbracket \subseteq \llbracket \psi \rrbracket$.

Based on the notion of logical meaning in terms of sets of possible worlds, we can also tackle another otherwise extremely elusive concept, namely that of *informativity*, i.e., the amount of information given. We say that φ is at least as informative as ψ iff $\llbracket \varphi \rrbracket \subseteq \llbracket \psi \rrbracket$, and that φ is strictly more informative than ψ iff $\llbracket \varphi \rrbracket \subset \llbracket \psi \rrbracket$.

This notion of logical informativity is a relative one. It does not give an absolute measure of information for each formula, but only compares two formulas in terms of their relative informativity. It also only compares some formulas; in fact, the ordering induced by this definition of logical informativity is a *partial order*.¹⁶

4.5 Argument schemas & validity

The important concept of logical validity can be thought, at least for **PRO-PL**OG, of as an extension of the notion of logical entailment to more than two formulas. Concretely, we defined logical validity of so-called argument schemas. If $\varphi_1, \varphi_2, \dots, \varphi_n$ and ψ are formulas of propositional logic, then $\varphi_1, \varphi_2, \dots, \varphi_n / \psi$ is an **ARGUMENT SCHEMA**. Formulas $\varphi_1, \varphi_2, \dots, \varphi_n$ are the **PREMISES**, formula ψ is the **CONCLUSION**.

The argument schema $\varphi_1, \varphi_2, \dots, \varphi_n / \psi$ is **VALID** iff for all valuation functions V such that $V(\varphi_1) = V(\varphi_2) = \dots = V(\varphi_n) = 1$ it also holds that $V(\psi) = 1$. If valid, we write $\varphi_1, \varphi_2, \dots, \varphi_n \models \psi$.

We can use truth tables again to check whether any given argument schema is valid or not. Consider the argument schema: $p \vee q, \neg q / p$. A natural language analogue would be the following. We know or assume the following premisses:

The murderer is the butler or the gardener.

The gardener is not the murderer.

And the putative conclusion is:

The butler is the murderer.

We would like to find out whether the conclusion logically follows from the premisses. Table 7 tells us that it does: in all rows (=valuation functions or

¹⁶Probabilistic notions of information, e.g., as given by information-theory extend the notion of logical information to an absolute measure of information which yields a total order.

possible worlds) in which all the premisses are true, which are marked with an asterisk (*) in Table 7, the conclusion (in the last column) is also true.

p	q	$p \vee q$	$\neg q$	/	q
1	1	1	0		1
1	0	1	1	*	1
0	1	1	0		0
0	0	0	1		0

Table 7: Truth table for checking the validity of argument schema

Table 8 shows a more extensive example of the truth-table method for the argument schema $\neg p \rightarrow (q \wedge \neg r)$, $\neg q \rightarrow \neg r / p$, which is not valid. The table also introduces names for each row, in terms of (types of) possible worlds, so that we can conveniently say that a possible world like w_6 in which p and r are false and q is true is a counterexample to validity: both premisses are true, but the conclusion is false.

	p	q	r	$\neg p$	$\neg r$	$q \wedge \neg r$	$\neg p \rightarrow (q \wedge \neg r)$	$\neg q$	$\neg q \rightarrow \neg r$	/	p
w_1	1	1	1	0	0	0	1	0	1	*	1
w_2	1	1	0	0	1	1	1	0	1	*	1
w_3	1	0	1	0	0	0	1	1	0		1
w_4	1	0	0	0	1	0	1	1	1	*	1
w_5	0	1	1	1	0	0	0	0	1		0
w_6	0	1	0	1	1	1	1	0	1	*	0
w_7	0	0	1	1	0	0	0	1	0		0
w_8	0	0	0	1	1	0	0	1	1		0

Table 8: Example of the truth-table method applied to an argument schema that is not valid

4.6 Theorems of propositional logic

Based on the definitions we have seen so far, we can make prove illuminating facts about PROPLOG. Here are some examples.

Proposition 1. If φ is a tautology, then $\neg\varphi$ is a contradiction.

Proof. If φ is a tautology, then for all valuation functions V we have $V(\varphi) = 1$. For any valuation function V if $V(\varphi) = 1$, by definition also $V(\neg\varphi) = 0$. Therefore, $V(\neg\varphi) = 0$ for all V , which means that $\neg\varphi$ is a contradiction. \square

Proposition 2. $\varphi \rightarrow \psi$ is a tautology iff φ logically entails ψ .¹⁷

Proof. First assume that $\varphi \rightarrow \psi$ is a tautology. We need to show that it follows that φ logically entails ψ . Towards contradiction, let us assume that φ does not logically entail ψ . This can only be the case if there is at least one valuation function V for which $V(\varphi) = 1$ and $V(\psi) = 0$. For this valuation function, by definition of the meaning of \rightarrow , it holds that $V(\varphi \rightarrow \psi) = 0$. So, we contradict our assumption that $\varphi \rightarrow \psi$ is a tautology.

¹⁷Notice that this may seem trivial, but there are really two notions that are related by this proposition. One is the definition of the sentential connective \rightarrow , the other is a high-level notion of logical entailment defined on top of the semantics for *all* formulas.

Next, we assume that φ logically entails ψ . We need to show that $\varphi \rightarrow \psi$ is a tautology. If φ logically entails ψ , this means that there is no valuation function V for which $V(\varphi) = 1$ and $V(\psi) = 0$. But, from the definition of the semantics of \rightarrow , we know that the only possibility for $\varphi \rightarrow \psi$ to be false is if $V(\varphi) = 1$ and $V(\psi) = 0$. Since this is ruled out by our assumption, $\varphi \rightarrow \psi$ must be a tautology. \square

Exercise 6. Use the truth-table method to check whether the following argument schemas are valid:

- (i) $p \rightarrow q, p / q$ [modus ponens]
- (ii) $p \rightarrow q, q / p$ [affirmation of the consequent]
- (iii) $p \rightarrow q, \neg q / \neg p$ [modus tollens]
- (iv) $\neg(p \wedge q), \neg q / p$

Exercise 7. For each of the following claims, determine whether it is true or false. If it is true prove it. If it is false, give a refutation by counterexample.

- (i) If φ is a tautology and ψ is a contradiction, $\varphi \rightarrow \psi$ is a contradiction.
- (ii) If ψ is a tautology, $\varphi \rightarrow \psi$ is a tautology.
- (iii) If φ is a tautology, $\varphi \rightarrow \psi$ is a tautology.
- (iv) If $\varphi \vee \psi$ is a contradiction, φ is a contradiction.