1

Understanding Propositional Logic

Propositional logic is about reasoning with propositions. These are sentences that can be assigned a truth value: *true* or *false*. They are built from primitive statements, called *atomic propositions*, by using *propositional logical connectives*. The truth values propagate over all propositions through *truth tables* for the propositional connectives. In this chapter I explain how to understand propositions and compute their truth values, and how to reason using schemes of propositions called *propositional formulae*. I will formally capture the concept of *logically correct propositional reasoning* by means of the fundamental notion of *propositional logical consequence*.

1.1 Propositions and logical connectives: truth tables and tautologies

1.1.1 Propositions

The basic concept of propositional logic is **proposition**. A proposition is a sentence that can be assigned a unique **truth value**: `true` or `false`.

Some simple examples of propositions include:

- The Sun is hot.
- The Earth is made of cheese.
- 2 plus 2 equals 22.
- The 1000th decimal digit of the number π is 9.
  (You probably don't know whether the latter is true or false, but it is surely *either true or false*.)

The following are not propositions (why?):

- Are you bored?
- Please, don't go away!
- She loves me.
- x is an integer.
- This sentence is false.

Logic as a Tool: A Guide to Formal Logical Reasoning, First Edition. Valentin Goranko.

Here is why. The first sentence above is a question, and it does not make sense to declare it true or false. Likewise for the imperative second sentence. The truth of the third sentence depends on who "she" is and who utters the sentence. Likewise, the truth of the fourth sentence is not determined as long as the variable x is not assigned a value, integer or not. As for the last sentence, the reason is trickier: assuming that it is true it truly claims that it is false – a contradiction; assuming that it is false, it falsely claims that it is false, hence it is not false – a contradiction again. Therefore, no truth value can be consistently ascribed to it. Such sentences are known as *self-referential* and are the main source of various *logical paradoxes* (see the appetizer and Russell's paradox in Section 5.2.1).

1.1.2 Propositional logical connectives

The propositions above are very simple. They have no logical structure, so we call them **primitive** or **atomic** propositions. From primitive propositions one can construct **compound** propositions by using special words called **logical connectives**. The most commonly used connectives are:

- not, called **negation**, denoted $\neg$;
- and, called **conjunction**, denoted $\wedge$ (or sometimes $\&$);
- or, called **disjunction**, denoted $\vee$;
- if ... then ..., called **implication**, or **conditional**, denoted $\rightarrow$;
- ... if and only if ..., called **biconditional**, denoted $\leftrightarrow$.

Remark 1 *It is often not grammatically correct to read compound propositions by simply inserting the names of the logical connectives in between the atomic components. A typical problem arises with the negation: one does not say "Not the Earth is square." A uniform way to get around that difficulty and negate a proposition P is to say "It is not the case that P."*

In natural language grammar the binary propositional connectives, plus others like *but, because, unless, although, so, yet*, etc. are all called "conjunctions" because they "conjoin", that is, connect, sentences. In logic we use the propositional connectives to connect propositions. For instance, given the propositions

"Two plus two equals five" and "The Sun is hot"

we can form the propositions

- "It is **not** the case that two plus two equals five. "
- "Two plus two equals five **and** the Sun is hot."
- "Two plus two equals five **or** the Sun is hot."
- "**If** two plus two equals five **then** the Sun is hot."
- "Two plus two equals five **if and only if** the Sun is hot."

For a more involved example, from the propositions (we assume we have already decided the truth value of each)

> "Logic is fun", "Logic is easy", and "Logic is boring"

we can compose a proposition

> "Logic is not easy or if logic is fun then logic is easy and logic is not boring."

It sounds better smoothed out a bit:

> "Logic is not easy or if logic is fun then it is easy and not boring."

1.1.3 Truth tables

How about the truth value of a compound proposition? It can be *computed* from the truth values of the components[1] by following the rules of 'propositional arithmetic':

- *The proposition* $\neg A$ *is true if and only if the proposition* A *is false.*
- *The proposition* $A \wedge B$ *is true if and only if both* A *and* B *are true.*
- *The proposition* $A \vee B$ *is true if and only if either of* A *or* B *(possibly both) is true.*
- *The proposition* $A \rightarrow B$ *is true if and only if* A *is false or* B *is true, that is, if the truth of* A *implies the truth of* B.
- *The proposition* $A \leftrightarrow B$ *is true if and only if* A *and* B *have the same truth values.*

We can systematize these rules in something similar to multiplication tables. For that purpose, and to make it easier for symbolic (i.e., mathematical) manipulations, we introduce a special notation for the two truth values by denoting the value `true` by `T` and the value `false` by `F`. Another common notation, particularly in computer science, is to denote `true` by **1** and `false` by **0**.

The rules of the "propositional arithmetic" can be summarized by means of the following **truth tables** (p and q below represent arbitrary propositions):

p	$\neg p$
T	F
F	T

p	q	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

[1] Much in the same way as we can compute the value of the algebraic expression $a \times (b - c) + b/a$ as soon as we know the values of a, b, c.

1.1.4 The meaning of the connectives in natural language and in logic

The use and meaning of the logical connectives in natural language does not always match their formal logical meaning. For instance, quite often the conjunction is loaded with a temporal succession and causal relationship that makes the common sense meanings of the sentences "The kid threw the stone and the window broke" and "The window broke and the kid threw the stone" quite different, while they have the same truth value by the truth table of the conjunction. Conjunction in natural language is therefore often non-commutative, while the logical conjunction is commutative. The conjunction is also often used to connect not entire sentences but only parts, in order to avoid repetition. For instance "The little princess is clever and beautiful" logically means "The little princess is clever and the little princess is beautiful." Several other conjunctive words in natural language, such as *but, yet, although, whereas, while* etc., translate into propositional logic as logical conjunction.

The disjunction in natural language also has its peculiarities. As for the conjunction, it is often used in a form which does not match the logical syntax, as in "The old stranger looked drunk, insane, or completely lost". Moreover, it is also used in an *exclusive* sense, for example in "I shall win or I shall die", while in formal logic we use it by convention in an *inclusive* sense, so "You will win or I will win" will be true if we both win. However, "exclusive or", abbreviated *Xor*, is sometimes used, especially in computer science. A few other conjunctive words in natural language, such as *unless*, can translate into propositional logic as logical disjunction, for instance "I will win, unless I die." However, it can also equivalently translate as an implication: "I will win, if I do not die."

Among all logical connectives, however, the implication seems to be the most debatable. Indeed, it is not so easy to accept that a proposition such as "If 2+2=5, then the Moon is made of cheese", if it makes any sense at all, should be assumed true. Even more questionable seems the truth of the proposition "If the Moon is made of chocolate then the Moon is made of cheese." The leading motivation to define the truth behavior of the implication is, of course, the logical meaning we assign to it. The proposition $A \to B$ means:

If *A is true*, ***then*** *B must be true*,

Note that if A is not true, then the (truth of the) implication $A \to B$ requires *nothing* regarding the truth of B. There is therefore only one case where that proposition should be regarded as false, namely when A is true, and yet B is not true. In all other cases we have no reason to consider it false. For it to be a proposition, it must be regarded true. This argument justifies the truth table of the implication. It is very important to understand the idea behind that truth table, because the implication is the logical connective which is most closely related to the concepts of logical reasoning and deduction.

Remark 2 *It helps to think of an implication as a promise. For instance, Johnnie's father tells him: "**If** you pass your logic exam, **then** I'll buy you a motorbike." Then consider the four possible situations: Johnnie passes or fails his exam and his father buys or does not buy him a motorbike. Now, see in which of them the promise is kept (the implication is true) and in which it is broken (the implication is false).*

Some terminology: the proposition A in the implication $A \to B$ is called the **antecedent** and the proposition B is the **consequent** of the implication.

The implication $A \to B$ can be expressed in many different but "logically equivalent" (to be defined later) ways, which one should be able to recognize:

- A implies B.
- B follows from A.
- If A, B.
- B if A.
- A only if B.
- B whenever A.
- A is sufficient for B.
(*Meaning: The truth of A is sufficient for the truth of B.*)
- B is necessary for A.
(*Meaning: The truth of B is necessary for A to be true.*)

1.1.5 Computing truth values of propositions

It can be seen from the truth tables that the truth value of a compound proposition does not depend on the meaning of the component propositions, but only on their truth values. To check the truth of such a proposition, we merely need to replace all component propositions by their respective truth values and then "compute" the truth of the whole proposition using the truth tables of the logical connectives. It therefore follows that

- "It is not the case that two plus two equals five" is true;
- "Two plus two equals five and the Sun is hot" is false;
- "Two plus two equals five or the Sun is hot" is true; and
- "If two plus two equals five, then the Sun is hot" is true (even though it does not make good sense).

For the other example, suppose we agree that

"Logic is fun" is true,
"Logic is boring" is false,
"Logic is easy" is true.

Then the truth value of the compound proposition

"Logic is not easy or if logic is fun then it is easy and not boring."

can be determined just as easily. However, in order to do so, we first have to analyze the *syntactic structure* of the proposition, that is, to determine how it has been composed, in other words in what order the logical connectives occurring therein have been applied. With algebraic expressions such as $a \times (b - c) + b/c$ that analysis is a little easier, thanks to the use of parentheses and the established priority order among the arithmetic operations. We also make use of parentheses and rewrite the sentence in the way (presumably) we all understand it:

"(Logic is not easy) or ((if logic is fun) then ((logic is easy) and (logic is not boring)))."

The structure of the sentence should be clear now. We can however go one step further and make it look exactly like an algebraic expression by using letters to denote the occurring primitive propositions. For example, let us denote

"Logic is fun" A,
"Logic is boring" B, and
"Logic is easy" C.

Now our compound proposition can be neatly rewritten as

$$(\neg C) \vee (A \rightarrow (C \wedge \neg B)).$$

In our rather informal exposition we will not use parentheses very systematically, but only whenever necessary to avoid ambiguity. For that purpose we will, like in arithmetic, impose a priority order among the logical connectives, namely:

- the negation has the strongest binding power, that is, the highest priority;
- then come the conjunction and disjunction;
- then the implication; and
- the biconditional has the lowest priority.

Example 3 *The proposition* $\neg A \vee C \rightarrow A \wedge \neg B$ *is a simplified version of* $((\neg A) \vee C) \rightarrow (A \wedge \neg B)$.

The last step is to compute the truth value. Recall that is not the actual meaning of the component propositions that matters but *only their truth values*, so we can simply replace the atomic propositions A, B, and C by their truth values and perform the formal computation following the truth tables step-by-step:

$$(\neg \mathtt{T}) \vee (\mathtt{T} \rightarrow (\mathtt{T} \wedge \neg \mathtt{F})) = \mathtt{F} \vee (\mathtt{T} \rightarrow (\mathtt{T} \wedge \mathtt{T})) = \mathtt{F} \vee (\mathtt{T} \rightarrow \mathtt{T}) = \mathtt{F} \vee \mathtt{T} = \mathtt{T}.$$

So, logic *is* easy after all! (At least, so far.)

1.1.6 Propositional formulae and their truth tables

If we only discuss particular propositions our study of logic would be no more useful than a study of algebra based on particular equalities such as $2 + 3 = 5$ or $12345679 \times 9 = 111111111$. Instead, we should look at *schemes of propositions* and their properties, just like we study algebraic formulae and equations and their properties. We call such schemes of propositions **propositional formulae**.

1.1.6.1 Propositional formulae: basics

I first define a **formal language** in which propositional formulae, meant to be templates for composite propositions, will be special words. That language involves:

- **propositional constants**: special fixed propositions $\top$, that always takes a truth value `true`, and $\bot$, that always takes a value `false`;

- **propositional variables** $p, q, r \ldots$, possibly indexed, to denote unspecified propositions in the same way as we use algebraic variables to denote unspecified or unknown numbers;
- the **logical connectives** that we already know; and
- **auxiliary symbols**: parentheses (and) are used to indicate the order of application of logical connectives and make the formulae unambiguous.

Using these symbols we can construct propositional formulae in the same way in which we construct algebraic expressions from variables and arithmetic operations. Here are a few examples of propositional formulae:

$$\top, p, \neg\bot, \neg\neg p, p \vee \neg q, \quad p_1 \wedge \neg(p_2 \to (\neg p_1 \wedge \bot))$$

There are infinitely many possible propositional formulae so we cannot list them all here. However, there is a simple and elegant way to give a precise definition of propositional formulae, namely the so-called **inductive definition** (or **recursive definition**). It consists of the following clauses or **formation rules**:

1. Every propositional constant or variable is a propositional formula.
2. If A is a propositional formula then $\neg A$ is a propositional formula.
3. If A, B are propositional formulae then each of $(A \vee B)$, $(A \wedge B)$, $(A \to B)$, and $(A \leftrightarrow B)$ is a propositional formula.

We say that a propositional formula is any string of symbols that can be constructed by applying – in some order and possibly repeatedly – the rules above, and only objects that can be constructed in such a way are propositional formulae.

Note that the notion of propositional formula that we define above is used in its own definition; this is the idea of *structural induction*. The definition works as follows: the first rule above gives us some initial stock of propositional formulae; as we keep applying the other rules, we construct more and more formulae and use them further in the definition. Eventually, every propositional formula can be obtained in several (finitely many!) steps of applying these rules. We can therefore think of the definition above as a construction manual prescribing how new objects (here, propositional formulae) can be built from already constructed objects. I discuss inductive definitions in more detail in Section 1.4.5.

From this point, I omit the unnecessary pairs of parentheses according to our earlier convention whenever that would not lead to syntactic ambiguity.

The formulae that are used in the process of the construction of a formula A are called **subformulae of** A. The last propositional connective introduced in the construction of A is called the **main connective of** A and the formula(e) to which it is applied is/are the **main subformula(e) of** A. I make all these more precise in what follows.

Example 4 (Construction sequence, subformulae and main connectives) *One construction sequence for the formula*

$$(p \vee \neg(q \wedge \neg r)) \to \neg\neg r$$

is

$$p, q, r, \neg r, \neg\neg r, q \wedge \neg r, \neg(q \wedge \neg r), p \vee \neg(q \wedge \neg r), (p \vee \neg(q \wedge \neg r)) \to \neg\neg r$$

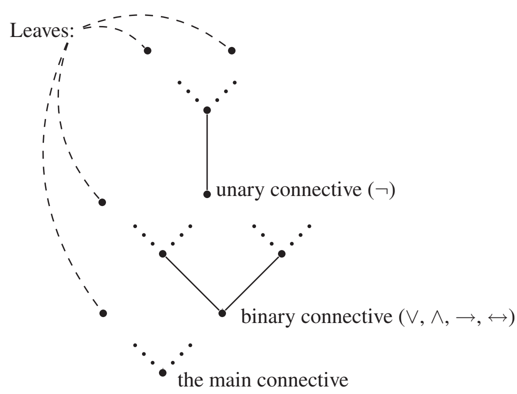
For instance, the subformula $(q \wedge \neg r)$ has main connective (the only occurrence of) $\wedge$ in it, and its main subformulae are q land $\neg r$; the first occurrence of $\neg$ is the main connective of $\neg(q \wedge \neg r)$ and its only main subformula is $(q \wedge \neg r)$; and the only occurrence of $\rightarrow$ is the main connective of the whole formula, the main subformulae of which are $(p \vee \neg(q \wedge \neg r))$ and $\neg\neg r$.

1.1.6.2 Construction tree and parsing tree of a formula

A sequence of formulae constructed in the process of applying the definition and ending with A is called a **construction sequence of a formula** A. A formula has many construction sequences and a construction sequence may contain many redundant formulae. A better notion for capturing the construction of a formula is the **construction tree** of that formula. A construction tree is a tree-like directed graph with nodes labeled with propositional constants, variables, and propositional connectives, such that:

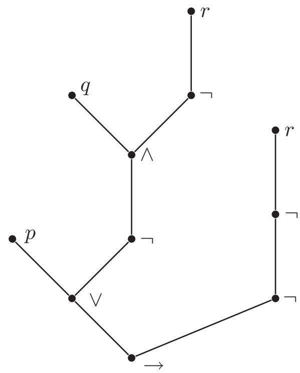
1. Every leaf is labeled by a propositional constant or variable.
2. Propositional constants and variables label only leaves.
3. Every node labeled with $\neg$ has exactly one successor node.
4. Every node labeled with any of $\wedge, \vee, \rightarrow$ or $\leftrightarrow$ has exactly two successor nodes: *left* and *right* successor.

A construction tree therefore looks like:



Every construction tree defines a formula C, built starting from the leaves and going towards the root, by applying at every node the formula construction rule corresponding to the label at that node. The formulae constructed in the process are precisely the subformulae of C, and the propositional connective labeling the root of the construction tree of a formula C is the main connective of C.

Example 5 (Construction tree) *The formula $(p \vee \neg(q \wedge \neg r)) \rightarrow \neg\neg r$ has the following construction tree:*



The **parsing tree** of a formula looks the same as its construction tree but is produced in inverse order, starting from the main connective (if any), drawing edges to the main components, and then recursively producing the parsing trees for each of them.

1.1.6.3 Truth assignments: satisfaction of propositional formulae

A propositional formula is a scheme that becomes a proposition whenever we substitute propositions for all occurring propositional variables. I, of course, mean **uniform substitutions**, that is, the same variables are replaced by the same propositions throughout the formula.

We cannot attribute a truth value to a propositional formula before we assign concrete propositions to all occurring propositional variables, for the same reason that we cannot evaluate $x(y+z)$ before we have assigned values to x, y, z. However, remember that in order to evaluate the truth value of a compound proposition, we only need to know the *truth values* of the occurring atomic propositions and not the propositions themselves.

For instance, if we substitute the propositions "0.5 is an integer" for p, "2 is less than 3" for q, and "the Moon is larger than the Sun" for r in the propositional formula

$$(p \vee \neg(q \wedge \neg r)) \rightarrow \neg\neg r$$

we find that the resulting proposition is true:

$$(\mathtt{F} \vee \neg(\mathtt{T} \wedge \neg\mathtt{F})) \rightarrow \neg\neg\mathtt{F} = (\mathtt{F} \vee \neg(\mathtt{T} \wedge \mathtt{T})) \rightarrow \neg\mathtt{T} = (\mathtt{F} \vee \neg\mathtt{T}) \rightarrow$$
$$\mathtt{F} = (\mathtt{F} \vee \mathtt{F}) \rightarrow \mathtt{F} = \mathtt{F} \rightarrow \mathtt{F} = \mathtt{T}.$$

If, however, we substitute *any* true propositions for p and q and a false proposition for r, then the resulting proposition will be false:

$$(\mathtt{T} \vee \neg(\mathtt{T} \wedge \neg\mathtt{F})) \rightarrow \neg\neg\mathtt{F} = (\mathtt{T} \vee \neg(\mathtt{T} \wedge \mathtt{T})) \rightarrow \neg\mathtt{T} = (\mathtt{T} \vee \neg\mathtt{T}) \rightarrow$$
$$\mathtt{F} = (\mathtt{T} \vee \mathtt{F}) \rightarrow \mathtt{F} = \mathtt{T} \rightarrow \mathtt{F} = \mathtt{F}.$$

Definition 6 *A function that assigns truth values to propositional variables in a given set is called **truth assignment** for that set of variables. If a truth assignment τ renders a formula A true, we say that τ **satisfies** A, denoted $\tau \models A$. A propositional formula is **satisfiable** if it is satisfied by some truth assignment.*

For instance, the formula $p \wedge \neg q \wedge r$ is satisfiable by the assignment p : T, q : F, r : T, while the formula $p \wedge \neg p$ is not satisfiable.

1.1.6.4 Truth tables of propositional formulae

Clearly, the truth of a given propositional formula only depends on the truth values assigned to the variables occurring in that formula. We can therefore think of propositional formulae as functions from truth assignments to truth values. We can tabulate the "behavior" of any propositional formula in a truth table where we list all possible truth assignments of the occurring variables, and for each of them we compute the corresponding truth value of the formula. We can do that by successively computing the truth values of all occurring subformulae, as we did just now. For example, the truth table for the above formula is compiled as follows:

p	q	r	$\neg r$	$\neg\neg r$	$q \wedge \neg r$	$\neg(q \wedge \neg r)$	$p \vee \neg(q \wedge \neg r)$	$(p \vee \neg(q \wedge \neg r)) \to \neg\neg r$
T	T	T	F	T	F	T	T	T
T	T	F	T	F	T	F	T	F
T	F	T	F	T	F	T	T	T
T	F	F	T	F	F	T	T	F
F	T	T	F	T	F	T	T	T
F	T	F	T	F	T	F	F	T
F	F	T						
F	F	F						

Exercise 7 *Complete the last two rows.*

Truth tables can be somewhat simplified if we notice that every occurrence of a logical connective in a propositional formula determines a *unique subformula* where that occurrence is the main connective of that subformula. We can now simplify the truth table of the formula by listing only the occurring variables and the whole formula, and then computing the truth table of every subformula in the column below the corresponding main connective:

p	q	r	$(p$	$\vee$	$\neg$	$(q$	$\wedge$	$\neg$	$r))$	$\to$	$\neg$	$\neg$	r
T	T	T	T	T	T	T	F	F	T	**T**	T	F	T
T	T	F	T	T	F	T	T	T	F	**F**	F	T	F
T	F	T	T	T	T	F	F	F	T	**T**	T	F	T
T	F	F	T	T	T	F	F	T	F	**F**	F	T	F
F	T	T	F	T	T	T	F	F	T	**T**	T	F	T
F	T	F	F	F	F	T	T	T	F	**T**	F	T	F
F	F	T								...			
F	F	F								...			

We therefore see that a propositional formula can represent a true or a false proposition, depending of the choice of the propositions substituted for the occurring propositional variables, or rather on their truth values.

1.1.7 Tautologies

Definition 8 *A propositional formula A is a* ***tautology*** *if it obtains a truth value* `T` *for* any *assignment of truth values to the variables occurring in A.*

The claim that A is a tautology is denoted

$$\models A.$$

Tautologies are also called ***(logically) valid formulae****.*

Thus, tautology always renders a true proposition; it represents a **logical law** in the same way as the identity $x + y = y + x$ represents a law of arithmetic and, therefore, holds no matter what values we assign to x and y.

Here are a few simple tautologies that represent some important features of propositional logic:

- $(p \vee \neg p)$: the **law of excluded middle**, which states that every proposition p is either true or false (and, in the latter case, $\neg p$ must be true);
- $\neg(p \wedge \neg p)$: the **law of non-contradiction**, which states that it cannot be the case that both p and $\neg p$ are true;
- $(p \wedge q) \rightarrow p$: this is always true by the very meaning (and truth table) of $\wedge$;
- likewise, $p \rightarrow (p \vee q)$ is always true by the very meaning and truth table of $\vee$; and
- $((p \wedge (p \rightarrow q)) \rightarrow q)$: if p is true and it is true that p implies q, then q is true. This reflects the meaning of the implication.

1.1.7.1 Checking tautologies with truth tables

How can we determine if a propositional formula A is a tautology? Quite easily: complete the truth table of A and check if it always takes a truth value `true`. Let us check some of the examples mentioned above:

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

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

The opposite concept of a tautology is a **contradictory formula**, that is, a formula that always takes a truth value `false`. For example, $(p \wedge \neg p)$ is a contradictory formula. A formula that is not contradictory is called **falsifiable**.

How are the concepts of tautology and contradictory formula, and the concepts of satisfiable and falsifiable formula related?

1. A formula A is a tautology precisely when its negation $\neg A$ is a contradictory formula, and A is contradictory precisely when its negation $\neg A$ is a tautology.
2. A formula A is satisfiable precisely when its negation $\neg A$ is falsifiable, and A is falsifiable precisely when its negation $\neg A$ is satisfiable.

1.1.7.2 Checking tautologies by searching for falsifying assignments

Checking tautologies with truth tables is straightforward but rather laborious and – let's admit – not very exciting. There is a somewhat streamlined and more intelligent method whereby we attempt to show that the formula *is not* a tautology by searching for an appropriate **falsifying truth assignment**, that is, a combination of truth values of the propositional variables that renders it false. If we succeed in finding such an assignment, then the formula is indeed not a tautology. If, however, when exploring a possible case we reach a state where some variable is required to be both true and false, that is clearly a *contradiction*, meaning that the case we are exploring is impossible and we must abandon that case. If *all* possible attempts to produce a falsifying assignment end up with a contradiction, then we have actually proved that the formula cannot be falsified; it must therefore be a tautology.[2]

The systematic search for a falsifying assignment is based on a step-by-step decomposition of the formula by using the truth tables of the propositional connectives occurring in it. Let us see how this works on some examples.

To make it more succinct I will use **signed formulae**, that is, expressions of the type $A : \mathtt{T}$, meaning "*A must be true*", and $A : \mathtt{F}$, meaning "*A must be false*".

1. Consider the formula $\neg(p \to \neg q) \to (p \vee \neg r)$.
   To falsify it, it must be the case that $\neg(p \to \neg q) : \mathtt{T}$ and $p \vee \neg r : \mathtt{F}$.
   For the former, it must be the case that $p \to \neg q : \mathtt{F}$, hence $p : \mathtt{T}$ and $\neg q : \mathtt{F}$.
   For the latter, it must be the case that $p : \mathtt{F}$ and $\neg r : \mathtt{F}$.
   This implies that p must be both true and false, which is impossible. Our attempt to falsify the formula has failed, and it is therefore a tautology.
2. Consider now $\neg p \to \neg(p \vee \neg q)$.
   To falsify it, it must be the case that $\neg p : \mathtt{T}$ while $\neg(p \vee \neg q) : \mathtt{F}$.
   Therefore $p : \mathtt{F}$ and $p \vee \neg q : \mathtt{T}$.
   For the latter, $p : \mathtt{T}$ or $\neg q : \mathtt{T}$. Let us consider both cases:
   Case 1: $p : \mathtt{T}$. This contradicts $p : \mathtt{F}$.
   Case 2: $\neg q : \mathtt{T}$. Then $q : \mathtt{F}$. In this case we have not reached any contradiction and there is nothing more we can do in order to obtain one. Indeed, we can check that $p : \mathtt{F}$ and $q : \mathtt{F}$ renders the formula false.
3. A contradiction in the truth values can be reached on any *subformula*, not necessarily a variable. For instance, take $(p \vee \neg q) \to (\neg p \to (p \vee \neg q))$. For it to be false, $(p \vee \neg q) : \mathtt{T}$

[2] This method of proof is called a *proof by contradiction* and is discussed in more detail in Section 5.1.

and $(\neg p \to (p \vee \neg q))$: F must be the case. From the latter, $\neg p$: T and $(p \vee \neg q)$: F, which contradicts the former.

4. Finally, take $((p \wedge \neg q) \to \neg r) \leftrightarrow ((p \wedge r) \to q)$. For it to be false, there are two possible cases:

   Case 1: $((p \wedge \neg q) \to \neg r)$: T and $((p \wedge r) \to q)$: F. The latter implies $(p \wedge r)$: T and q : F, hence p : T, q : F, r : T. For the former, two sub-cases are possible:
   Case 1a: $\neg r$: T. Then r : F, which is a contradiction with r : T.
   Case 1b: $(p \wedge \neg q)$: F. Then:
   Case 1bi: p : F, a contradiction with p : T.
   Case 1bii: $\neg q$: F and q : T, again a contradiction but now with q : F.
   Note that the consideration of these sub-cases could have been avoided, if we had noticed that the assignment p : T, q : F, r : T renders $((p \wedge \neg q) \to \neg r)$: F.

   Case 2: $(((p \wedge r) \to q)$: T and $(p \wedge \neg q) \to \neg r)$: F. The former implies $(p \wedge \neg q)$: T and $\neg r$: F, that is, p : T, q : F, r : T. Again, we can either notice that this assignment renders $((p \wedge r) \to q)$: F, or consider the cases for $((p \wedge r) \to q)$: T and see that all lead to a contradiction.
   The formula cannot be falsified in either case, so it is a tautology.

This method can be formalized and mechanized completely into a kind of *deductive system*, called Semantic Tableaux, which is presented in Section 2.3.

References for further reading

For helpful and accessible introductions to propositional logic and discussions of the issues covered here plus more, see Tarski (1965), Gamut (1991), Jeffrey (1994), Barwise and Echemedy (1999), Nederpelt and Kamareddine (2004), Boole (2005), Bornat (2005), Hodges (2005), Chiswell and Hodges (2007), Makinson (2008), Ben-Ari (2012), and van Benthem *et al*. (2014).

Some suitable books on philosophical logic, old and new, include Carroll (1897), Kalish and Montague (1980), Smith (2003), Copi *et al*. (2010), and Halbach (2010).

For more technical books on mathematical logic, see Shoenfield (1967), van Dalen (1983), Hamilton (1988), Mendelson (1997), Enderton (2001), and Hedman (2004),

For books on computational and applied logic the reader is referred to Gallier (1986), Nerode and Shore (1993), and Fitting (1996).

Last but not least, some fun logic books include Carroll (1886) and Smullyan (1998, 2009a, b, 2011, 2013, 2014).

Exercises

1.1.1 Which of the following are propositions? (Assume that John, Mary and Eva are concrete individuals.)

(a) $2^3 + 3^2 = 19$	(c) $2^3 + 3^2 = x$
(b) $2^3 + 3^2 = 91$	(d) Will you marry me?

(e) John married on 1 January 1999.
(f) John must marry Mary!
(g) I told her about John.
(h) Mary is not happy if John married Eva.
(i) Who is Eva?
(j) Mary is not happy if $2^3 + 3^2 = 19$.
(k) This sentence refers to itself.
(l) This sentence is true.
(m) If you are reading this sentence now, then it is not true.

1.1.2 If A and B are true propositions and C and D are false propositions, determine the truth values of the following compound propositions without using truth tables.

(a) $A \wedge (B \vee C)$
(b) $(C \to A) \to D$
(c) $C \to (A \to D)$
(d) $\neg(\neg A \vee C) \wedge B$
(e) $\neg(\neg(\neg D \wedge (B \to \neg A)))$
(f) $\neg(C \to A) \vee (C \to D)$
(g) $(C \leftrightarrow \neg B) \vee (A \to \neg A)$
(h) $(A \leftrightarrow \neg B) \leftrightarrow (C \leftrightarrow \neg D)$

1.1.3 Determine the antecedent and the consequent in each of the following implications.

(a) Whenever John talks everyone else listens.
(b) Everyone else listens if John talks.
(c) John talks only if everyone else listens.
(d) If everyone else listens, John talks.
(e) An integer is positive if its cube is positive.
(Hint: To make it easier to reason, introduce a name for the object in question, for example "An integer n is positive if the cube of n is positive.")
(f) An integer is positive only if its cube is positive.
(g) A function is continuous whenever it is differentiable.
(h) The continuity of a function is necessary for it to be differentiable.
(i) The continuity of a function is sufficient for it to be differentiable.

1.1.4 A positive integer n is called **prime** if $n > 1$ and n is divisible only by 1 and by itself. Which of the following conditions are sufficient and which are necessary for the truth of "n is not prime", where n is some (given) positive integer?

(a) n is divisible by 3.
(b) n is even.
(c) n is divisible by 6.
(d) n has at least two different factors.
(e) n has more than two different factors.
(f) $n = 15$.
(g) n has a factor different from n.
(h) n has a prime factor.
(i) n has a prime factor different from n.

1.1.5 Write each of the following composite propositions in a symbolic form by identifying its atomic propositions and logical structure. Then determine its truth value.

(a) The Earth rotates around itself and, if the Moon rotates around the Earth, then the Sun rotates around the Moon.
(b) If the Sun rotates around the Earth or the Earth rotates around the Moon then the Sun rotates around the Moon.
(c) The Moon does not rotate around the Earth if the Sun does not rotate around the Earth and the Earth does not rotate around the Moon.

(d) The Earth rotates around itself only if the Sun rotates around the Earth or the Moon does not rotate around the Earth.
(e) The Earth rotates around itself if and only if the Moon does not rotate around the Earth or the Earth does not rotate around the Moon.

1.1.6 Determine the truth value of the proposition A in each of the following cases, without using truth tables. (Hint: if necessary, consider the possible cases.)
(a) B and $B \to A$ are true.
(b) $A \to B$ is true and B is false.
(c) $\neg B$ and $A \vee B$ are true.
(d) Each of $B \to \neg A$, $\neg B \to \neg C$, and C is true.
(e) Each of $\neg C \wedge B$, $C \to (A \vee B)$, and $\neg(A \vee C) \to C$ is true.

1.1.7 Let P, Q, R, and S be propositions. Show that:
(a) If the propositions P and $P \to Q$ are true, then Q is true.
(b) If the propositions $(P \vee Q) \to R$ and $P \vee R$ are true, then R is true.
(c) If $P \to Q, Q \to R$, and $P \wedge S$ are true, then $R \wedge S$ is true.
(d) If $\neg P \to \neg Q, \neg(P \wedge \neg R), \neg R$ are true, then Q is false.
(e) If $P \to Q$, $R \vee (S \wedge \neg Q)$, and $\neg R$ are true, then P is false.
(f) If $Q \to (R \wedge S)$ is true and $Q \wedge S$ is false, then $R \wedge Q$ is false.
(g) If $P \to Q$ and $Q \to (R \vee S)$ are true and $P \to R$ is false, then $\neg R \to S$ is true.
(h) If $\neg P \to (\neg Q \vee \neg R)$ and $Q \wedge (P \vee R)$ are true, then P is true.
(i) If $P \to Q$ and $Q \to (R \vee S)$ are true and $P \to R$ is false, then S is true.
(j) If $Q \to (R \wedge S)$ is true and $Q \wedge S$ is false, then Q is false.

1.1.8 Construct the truth tables of the following propositional formulae, and determine which (if any) of them are tautologies and which are contradictory formulae.

(a) $\neg(p \to \neg p)$
(b) $p \vee (p \to \neg p)$
(c) $p \wedge (q \vee \neg q)$
(d) $(p \wedge \neg p) \to q$
(e) $((p \to q) \to p) \to p$
(f) $\neg p \wedge \neg(p \to q)$
(g) $(p \vee \neg q) \to \neg(q \wedge \neg p)$
(h) $(p \to q) \wedge (q \to r) \wedge \neg(\neg p \vee r)$
(i) $\neg(\neg p \leftrightarrow q) \wedge (r \vee \neg q)$
(j) $\neg((p \wedge \neg q) \to r) \leftrightarrow (\neg(q \vee r) \to \neg p)$

1.1.9 Determine which (if any) of the following propositional formulae are tautologies by searching for falsifying truth assignments.
(a) $q \to (q \to p)$
(b) $p \to (q \to p)$
(c) $((p \to q) \wedge (p \to \neg q)) \to \neg p$
(d) $((p \to q) \vee (p \to \neg q)) \to \neg p$
(e) $(p \vee \neg q) \wedge (q \to \neg(q \wedge \neg p))$
(f) $((p \to q) \wedge (q \to r)) \to (r \to p)$
(g) $((p \to q) \wedge (q \to r)) \to (p \to r)$
(h) $((p \to q) \wedge (p \to r)) \to (p \to (q \wedge r))$
(i) $((p \wedge q) \to r) \to ((p \to r) \wedge (q \to r))$
(j) $((p \vee q) \to r) \to ((p \to r) \wedge (q \to r))$

(k) $((\neg p \wedge q) \rightarrow \neg r) \rightarrow (\neg q \rightarrow \neg(p \wedge \neg r))$
(l) $((p \rightarrow r) \vee (q \rightarrow r)) \rightarrow ((p \vee q) \rightarrow r)$
(m) $((p \rightarrow r) \wedge (q \rightarrow r)) \rightarrow ((p \vee q) \rightarrow r)$
(n) $p \rightarrow ((q \rightarrow r) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r)))$

1.1.10 Lastly, some logical puzzles[3]. On the remote planet Nologic there are two types of intelligent creatures:

- *truth-tellers*, who always tell the truth, and
- *liars*, who (you guessed it) always lie.

It is not possible to distinguish them by appearance, but only by the truth or falsity of the statements they make.

(a) A space traveler visited Nologic and met two inhabitants, P and Q. He asked them: "Is any of you a liar?" "At least one of us is a liar", replied P. Can you find out what P and Q are?

(b) Next, the stranger met two other inhabitants and asked one of them "Is any of you a liar?". He got a "yes" or "no" answer and from that answer was able to determine for each of them whether he is a liar or not. What was the answer and what was the stranger's conclusion?

(c) Walking about Nologic, the stranger met two other inhabitants A and B and asked A, "Is any of you a truth-teller?" "If B is a liar, then I am a liar too", replied A. What are A and B?

(d) The stranger went on and met three locals X, Y, and Z and asked X: "Are you a liar?" X answered something which the stranger did not hear, so he asked Y: "What did X say?" "He said that he is a liar", replied Y. Then Z added "Don't believe Y, he is a liar". Can you identify all liars?

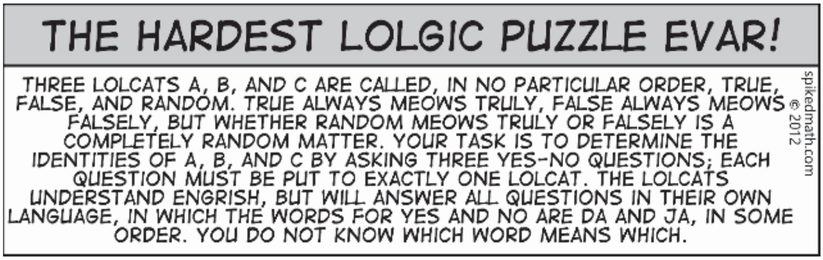
(e$*$) The stranger went on. In the evening, he began to look for a shelter for the night, but was very cautious because he knew that some of the inhabitants were man-eaters and it was not possible to recognize them by appearance. He met three inhabitants, C, D, and E. He asked C, "How many of you are truth-tellers?" "Flam flim" answered C in her language. "What did she say?" asked the stranger D. "Just one", replied D. "Do not trust D, he is a liar. Come with me, I'm not a man-eater" said E. "No, come with me, I'm not a man-eater" countered D.
What should the stranger do?

(f$*$) The stranger decided to go back to the port where his spaceship was, but he got lost. After a long walk he got to a fork of the road. He knew that one of the two roads would take him to the spaceship port, but did not know which one. Luckily, he saw two of the inhabitants, one on each road. He had met them before so he knew that one of them was a liar and the other a truth-teller, but could not remember who was who.

Can the stranger ask just one question to either of these inhabitants in order to find out which is the correct road to the spaceship port? If so, what question should he ask?

[3] For many more puzzles of this type, I warmly recommend to the reader the marvellous logical puzzle books by Raymond Smullyan (1998, 2009a, b, 2011, 2013, 2014).

1.1.11



The early origins of propositional logic

The Megarian school of philosophy

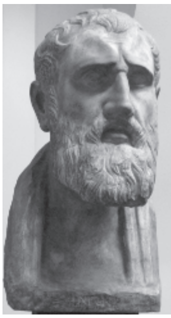
The **Megarian school** of philosophy was founded by **Euclid of Megara** (c. 430–360 BC). (This is not the famous geometer Euclid of Alexandria.) He was a disciple of Socrates and, following on his ideas, claimed that there is one universal Good in the world, sometimes also called Wisdom, God, or Reason, and that nothing that is not Good exists. Euclid used logic, in a dialogue form, to defend his ideas and win arguments. He applied extensively **reductio ad absurdum** (see Section 2.4) in his argumentation.

Eubulides (4th century BC) was a pupil of Euclid of Megara and a strong opponent of Aristotle. He was most famous for inventing several **paradoxes**, still boggling the mind today. The most popular of them is the *Liar's paradox*, also attributed to **Epimenides** (6th century BC), a Cretan, who is claimed to have said that " *All Cretans are liars*" (which is not a paradox yet, just a necessarily false statement).

Diodorus Cronus (?–c. 284 BC) was another prominent philosopher from the Megarian **Dialectical school**. He made important early contributions to logic, especially on the **theory of conditionals** and the concepts of "possible" and "necessary", thus following Aristotle in laying the foundations of **modal logic**. He is most famous for his **Master argument** in response to Aristotle's discussion of **future contingents**, such as "*There will be a sea battle tomorrow*". Diodorus' argument implied that whatever is possible is actually necessary, hence there are no contingencies.

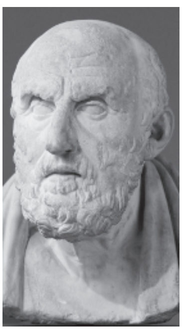
Philo of Megara (c. 400 BC) was a disciple of Diodorus Cronus. He was also his most famous opponent in their disputes concerning the modal notions of "possible" and "necessary" and on the criteria for truth of conditional statements. Notably, Philo regarded a conditional as false only if it has both a true antecedent and a false consequent, essentially inventing the **truth-functional implication** which we now use in classical propositional logic, also known as **material implication**.

The Stoic school of philosophy



Greek philosopher **Zeno of Citium** (c. 335–265 BC), a pupil of Diodorus Cronus, founded the **Stoic school** in the early 3rd century BC. Zeno and his school had an elaborated theory of philosophy as a way of life, and also made influential contributions to physics, cosmology, epistemology and ethics.

Zeno taught that the *Universal Reason* (*Logos*, from which the word *Logic* originated) was the greatest good in life and living in accordance with it was the purpose of human life. The Stoic school was engaged in logical argumentation and essentially laid the foundations of propositional logic as an alternative to the Aristotelian logic of Syllogisms (see Section 3.5).



Chrysippus (c. 280–207 BC) was a philosopher and logician from the Stoic School. He wrote over 300 books (very few of survived to be studied) on many fundamental topics of logic, including propositions and propositional connectives (he introduced the implication, conjunction and exclusive disjunction), logical consequence, valid arguments, logical deduction, causation, and logical paradoxes, and on the most popular **non-classical logics**, including modal, tense, and epistemic logics. Chrysippus is often regarded as the founder of propositional logic, and is one of the most important early formal logicians along with Aristotle.

1.2 Propositional logical consequence: logically correct inferences

The central problem of logic is the study of *correct argumentation and reasoning*. In this section I define and discuss what it means for an argument to be logically correct or not by formalizing the fundamental logical concept of *logical consequence*. This is done here just for a simple type of logical arguments that only involve propositional reasoning, called *propositional arguments*.

1.2.1 Propositional logical consequence

The intuition behind logically correct reasoning is simple: starting from true premises should always lead to true conclusions. Let us first make this intuition precise.

Definition 9 *A propositional formula B is a* **logical consequence** *of the propositional formulae $A_1, \ldots, A_n$, denoted*[4]

$$A_1, \ldots, A_n \models B$$

[4] Note that I use here the same symbol we used to indicate tautologies. This will be justified soon.

if B is true whenever all $A_1, \ldots, A_n$ are true. That means: if every truth assignment to the variables occurring in $A_1, \ldots, A_n, B$ for which the formulae $A_1, \ldots, A_n$ is true, then the formula B is also true.

*When $A_1, \ldots, A_n \models B$, we also say that B **follows logically from** $A_1, \ldots, A_n$, or that $A_1, \ldots, A_n$ **imply logically** B.*

In the context of $A_1, \ldots, A_n \models B$, the formulae $A_1, \ldots, A_n$ are called **assumptions** while the formula B is called a **conclusion**.

When $A_1, \ldots, A_n \models B$ is not the case, we write $A_1, \ldots, A_n \not\models B$.

If $A_1, \ldots, A_n \models B$ then every substitution of propositions for the variables occurring in $A_1, \ldots, A_n, B$ which turns the formulae $A_1, \ldots, A_n$ into true propositions also turns the formula B into a true proposition.

In order to check whether $A_1, \ldots, A_n \models B$ we can simply complete the truth tables of $A_1, \ldots, A_n, B$ and check, row by row, if the following holds: *whenever all formulae $A_1, \ldots, A_n$ have a truth value* T *in that row, B must also have a truth value* T. (Of course, it is possible for B to be true without any of $A_1, \ldots, A_n$ being true.) If that holds in *every row* in the table, then B *does* follow logically from $A_1, \ldots, A_n$; if that fails *in at least one row*, then B *does not* follow logically from $A_1, \ldots, A_n$.

Thus, B *does not* follow logically from $A_1, \ldots, A_n$, just in case there is a truth assignment which renders all formulae $A_1, \ldots, A_n$ true and B false.

Example 10 (Some simple cases of logical consequences)

1. *Any formula B follows logically from any set of formulae that contains B. (Why?)*
2. *Any tautology follows logically from any set of formulae,* even from the empty set!
3. *For any formulae P and Q we claim that $P, P \to Q \models Q$.*
   Note first that, whatever the formulae P and Q, any truth assignment eventually renders each of them true or false and all combinations of these truth values can be possible, so we can treat P and Q as propositional variables and consider the truth tables for P, Q and $P \to Q$:

P	Q	P	$P \to Q$	Q
T	T	T	T	T
T	F	T	F	F
F	T	F	T	T
F	F	F	T	F

   Indeed, in every row where the 3rd and 4th entries are T*, the 5th entry is also* T.
4. *$P \to R, Q \to R \models (P \vee Q) \to R$ for any formulae P, Q, R.*
   Likewise, it suffices to show that $p \to r, q \to r \models (p \vee q) \to r$, for propositional variables p, q, r.
   Indeed, in every row of the truth table where the truth values of $p \to r$ and $q \to r$ are T*, the truth value of $(p \vee q) \to r$ is also* T.

p	q	r	$p \to r$	$q \to r$	$p \vee q$	$(p \vee q) \to r$
T	T	T	T	T	T	T
T	T	F	F	F	T	F
T	F	T	T	T	T	T
T	F	F	F	T	T	F
F	T	T	T	T	T	T
F	T	F	T	F	T	F
F	F	T	T	T	F	T
F	F	F	T	T	F	T

5. *Is it true that* $p \vee q, q \to p \models p \wedge q$*? We check the truth table:*

p	q	$p \vee q$	$q \to p$	$p \wedge q$
T	T	T	T	T
T	F	T	T	F
F	T	...	...	...
F	F	...	...	...

and see that in the 2nd row both assumptions are true while the conclusion is false. This suffices to conclude that $p \vee q, q \to p \not\models p \wedge q$*, so there is no need to fill in the truth table any further.*

Recall that $\models A$ means that A is a tautology. Tautologies and logical consequences are closely related. First, note that a formula A is a tautology if and only if (iff) it follows logically from the empty set of formulae. Indeed, we have already noted that if $\models A$ then $\emptyset \models A$. Now, to see that if $\emptyset \models A$ then $\models A$, suppose $\emptyset \models A$ and take *any* truth assignment. Note that it satisfies *every formula* from $\emptyset$. Why? Well, there can be *no* formula in $\emptyset$ which is not satisfied, because there are no formulae in $\emptyset$ at all! Since $\emptyset \models A$, that truth assignment must also satisfy A. Thus, A is satisfied by every truth assignment[5].

In general, we have the following equivalences.

Proposition 11 *For any propositional formulae* $A_1, \ldots, A_n, B$*, the following are equivalent:*

1. $A_1, \ldots, A_n \models B$
2. $A_1 \wedge \cdots \wedge A_n \models B$
3. $\models (A_1 \wedge \cdots \wedge A_n) \to B$
4. $\models A_1 \to (\cdots \to (A_n \to B) \cdots)$

I leave the proofs of these equivalences as easy exercises.

Checking logical consequences can be streamlined, in the same way as checking tautologies, by organizing a systematic search for a **falsifying assignment**. In order to check

[5] Here we did some logical reasoning based on the very same concepts of truth and logical consequence that we are discussing. When reasoning about logic, this kind of bootstrapping reasoning is inevitable!

if B follows logically from $A_1, \ldots, A_n$ we look for a truth assignment to the variables occurring in $A_1, \ldots, A_n, B$ that renders all $A_1, \ldots, A_n$ true and B false. If we succeed, then we have proved that B *does not* follow logically from $A_1, \ldots, A_n$; otherwise we want to prove that no such assignment is possible by showing that the assumption that it exists leads to a contradiction.

For example, let us check again that $p \to r, q \to r \models (p \vee q) \to r$. Suppose that for some assignment $(p \to r) : \mathtt{T}$, $(q \to r) : \mathtt{T}$ and $((p \vee q) \to r) : \mathtt{F}$. Then $(p \vee q) : \mathtt{T}$ and $r : \mathtt{F}$, hence $p : \mathtt{T}$ or $q : \mathtt{T}$.

Case 1: $p : \mathtt{T}$. Then $(p \to r) : \mathtt{F}$, that is, a contradiction.
Case 2: $q : \mathtt{T}$. Then $(q \to r) : \mathtt{F}$, again, a contradiction.

Thus, there is no assignment that falsifies the logical consequence above.

1.2.2 Logically sound rules of propositional inference and logically correct propositional arguments

We now apply the notion of logical consequence to define and check whether a given propositional argument is logically correct. Let us first introduce some terminology.

Definition 12 *A **rule of propositional inference** (**inference rule**, for short) is a scheme:*

$$\frac{P_1, \ldots, P_n}{C}$$

*where $P_1, \ldots, P_n, C$ are propositional formulae. The formulae $P_1, \ldots, P_n$ are called **premises** of the inference rule, and C is its **conclusion**.*

*An **instance** of an inference rule is obtained by uniform substitution of concrete propositions for the variables occurring in all formulae of the rule. Every such instance is called a **propositional inference**, or a **propositional argument** based on that rule.*

Definition 13 *An inference rule is **(logically) sound** if its conclusion follows logically from the premises. A propositional argument is **logically correct** if it is an instance of a logically sound inference rule.*

Example 14

1. *The following inference rule*

$$\frac{p, p \to q}{q}$$

   *is sound, as we have already seen. This rule is known as the **Detachment rule** or **Modus Ponens**, and is very important in the logical deductive systems called* axiomatic systems *which we will study in Chapter 2.*
   The inference

   Alexis is singing.
   If Alexis is singing, then Alexis is happy.
   ____
   Alexis is happy.

   is therefore logically correct, being an instance of that inference rule.

2. *The inference rule*

$$\frac{p, q \to p}{q}$$

is not sound*: if p is true and q is false, then both premises are true while the conclusion is false.*
Therefore the inference

5 is greater than 2.
5 is greater than 2 if 5 is greater than 3.
---
5 is greater than 3.

is not logically correct, despite the truth of both premises and the conclusion, as it is an instance of an unsound rule.

Some remarks are in order here.

- It is very important to realize that the logical correctness of an inference *does not always guarantee the truth of the conclusion, but only when all premises are true*. In other words, if at least one premise of a logically correct inference is false, then the conclusion may also be false. For instance, the inference

  5 divides 6.
  If 5 divides 6, then 5 divides 11.
  ---
  5 divides 11.

  is logically correct (being an instance of the rule Modus Ponens) in spite of the falsity of the conclusion. This does not contradict the idea of logical correctness of an inference, because the first premise is false. (What about the second premise?)
- Conversely, if the conclusion of an inference happens to be true, this does not necessarily mean that the inference is logically correct as in the second example above.
- Moreover, it may happen that the truth of the premises of an inference *does imply* the truth of the conclusion, and yet the inference is not *logically* correct. For instance, the correctness of the inferences

  Today is Monday.

  Tomorrow will be Tuesday.

or

$$\frac{a = 2, a + b = 5}{b = 3}$$

is based *not* on logical consequence but, in the first case, on the commonly known fact that Tuesday always follows after Monday, and in the second case on some laws of arithmetic. Indeed, the first inference is based on the rule

$$\frac{p}{q}$$

and the second inference (although the statements are not really propositions) on

$$\frac{p, q}{r},$$

both of which are clearly unsound.

To summarize, the meaning and importance of logically correct inferences is that only such inferences guarantee that *if* all premises are true, *then* the conclusions will also be true. That is why only logically correct inferences are safe to be employed in our reasoning.

Let us now look at a few more examples.

1. The inference rule

$$\frac{q, p \vee \neg q}{p}$$

is logically sound. You can check this in two ways: by applying the definition or by showing that the corresponding formula

$$(q \wedge (p \vee \neg q)) \rightarrow p$$

is a tautology.

Consequently, the inference

Olivia is crying or Olivia is not awake.
Olivia is awake.
―――――――――――――
Olivia is crying.

is logically correct, being an instance of the rule above.

2. Now, take the argument

If a divides b or a divides c, then a divides bc.
a divides bc.
a does not divide b.
―――――――――――――
Therefore a divides c.

where a, b, c are certain integers. This argument is an instance of the following inference rule:

$$\frac{(p \vee q) \rightarrow r, r, \neg p}{q}.$$

Let us see if we can invalidate this rule. For that we need an assignment such that $((p \vee q) \rightarrow r) : \mathtt{T}$, $r : \mathtt{T}$, and $\neg p : \mathtt{T}$, hence $p : \mathtt{F}$ and $q : \mathtt{F}$. Indeed, the assignment $p : \mathtt{F}$, $q : \mathtt{F}$, and $r : \mathtt{T}$ renders all premises true and the conclusion false. (Check this.)

The rule is therefore not logically sound, hence the argument above is not logically correct.

I develop the method behind the last argument above in the next chapter.

1.2.3 Fallacies of the implication

As an application let us analyze some very common forms of correct and incorrect reasoning related to implications. Given the implication

$$A \rightarrow B$$

we can form the so-called **derivative implications**:

- the **converse** of $A \to B$ is $B \to A$;
- the **inverse** of $A \to B$ is $\neg A \to \neg B$; and
- the **contrapositive** of $A \to B$ is $\neg B \to \neg A$.

Now, suppose we know that $A \to B$ is true. What can we say about the truth of its derivatives? To answer that question, look at each of the inferences:

$$\frac{A \to B}{B \to A}, \qquad \frac{A \to B}{\neg A \to \neg B}, \qquad \frac{A \to B}{\neg B \to \neg A}$$

Exercise 15 *Show that the first two of these inferences are incorrect, while the third inference is correct.*

The truth of an implication $A \to B$ therefore *only implies the truth of its contrapositive*, but not the truth of the converse or inverse. These are mistakes that people often make, respectively called **the fallacy of the converse implication** and **the fallacy of the inverse implication**. For example, the truth of the implication "If it has just rained, then the tennis court is wet" does not imply that either of "If the tennis court is wet, then it has just rained" and "If it has not just rained, then the tennis court is not wet" is true – someone may have just watered the court on a clear sunny day – but it certainly implies that "If the court is not wet, then it has not just rained." In fact, it can easily be checked that

$$\frac{\neg B \to \neg A}{A \to B}$$

is also logically correct. This is the basis of the method of *proof by contraposition*, which is discussed further in Section 2.5.

References for further reading
Propositional logical consequence, as well as propositional arguments, inference rules and their logical correctness, are treated in more details in Carroll (1897), Tarski (1965), Kalish and Montague (1980), Gamut (1991), Nerode and Shore (1993), Jeffrey (1994), Barwise and Echemendy (1999), Smith (2003), Boole (2005), Bornat (2005), Chiswell and Hodges (2007), Copi *et al*. (2010), Halbach (2010), Ben-Ari (2012), and van Benthem *et al*. (2014).

Exercises

1.2.1 Prove Proposition 11. (Hint: you do not have to prove all pairs of equivalences. It is sufficient to show, for instance, that claim 1 implies 2, which implies 3, which implies 4, which implies 1.)

1.2.2 Show that the first two of the following inference rules, corresponding to the derivative implications, are logically unsound while the third rule is sound.

(a) $\dfrac{p \to q}{q \to p}$ (b) $\dfrac{p \to q}{\neg p \to \neg q}$ (c) $\dfrac{p \to q}{\neg q \to \neg p}$

1.2.3 Using truth tables, check if the following inference rules are sound.

(a) $\dfrac{p \to q, \neg q \vee r, \neg r}{\neg p}$

(b) $\dfrac{\neg p \to \neg q, q, \neg(p \wedge \neg r)}{r}$

(c) $\dfrac{p \to q, p \vee \neg r, \neg r}{\neg q \vee r}$

(d) $\dfrac{((p \wedge q) \to r), \neg(p \to r)}{q \to r}$

1.2.4 Write down the inference rules on which the following arguments are based and check their logical soundness, using truth tables.

(a) In the following argument, X is a certain number.

X is greater than 3.
_____
X is greater than or equal to 3.

(b) In the following argument, Y is a certain number.

If Y is greater than –1, then Y is greater than –2.
Y is not greater than –2.
_____
Y is not greater than –1.

(c)

If the triangle ABC has a right angle, then it is not equilateral.
The triangle ABC does not have a right angle.
_____
Therefore, the triangle ABC is equilateral.

(d)

If Victor is good at logic, then he is clever.
If Victor is clever, then he is rich.
_____
Therefore, if Victor is good at logic, then he is rich.

(e) In the following argument n is a certain integer.

If n is divisible by 2 and n is divisible by 3, then n is divisible by 6.
If n is divisible by 6, then n is divisible by 2.
n is not divisible by 3.
_____
Therefore, n is not divisible by 6.

1.2.5 For each of the following implications construct the converse, inverse and the contrapositive, phrased in the same way.

(a) If a is greater than –1, then a is greater than –2.
(b) x is not prime if x is divisible by 6.
(c) x is positive only if its square is positive.

(d) The triangle ABC is equilateral whenever its medicentre and orthocentre coincide.
(e) For the function f to be continuous, it is sufficient that it is differentiable.
(f) For a function not to be differentiable, it is sufficient that it is discontinuous.
(g) For the integer n to be prime, it is necessary that it is not divisible by 10.

George Boole (2.11.1815–8.12.1864) was an English mathematician who first proposed and developed an algebraic approach to the study of logical reasoning. Boole's first contribution to logic was a pamphlet called *Mathematical Analysis of Logic*, written in 1847. He published his main work on logic, *An Investigation of the Laws of Thought, on which are Founded the Mathematical Theories of Logic and Probabilities*, in 1854. In it Boole developed a general mathematical method of logical inference, laying the foundations of modern mathematical logic. His system proposed a formal algebraic treatment of propositions by processing only their two possible truth values: yes–no, true–false, zero–one. In Boole's system, if x stands for "white things" then $1 - x$ stands for "non-white things;" if y stands for "sheep", then xy stands for "white sheep", etc. $x(1 - x)$ denotes things that are both white and non-white, which is impossible. A proposition of the shape $x(1 - x)$ is therefore always false, that is, has a truth value 0. The algebraic law $x(1 - x) = 0$ therefore emerges. The resulting algebraic system is known today as (the simplest) **Boolean algebra**.

Boole also argued that symbolic logic is needed in other mathematical disciplines, especially in probability theory. He wrote: "$\cdots$ no general method for the solution of questions in the theory of probabilities can be established which does not explicitly recognise those universal laws of thought which are the basis of all reasoning $\cdots$"

Propositional logic today is somewhat different from Boole's system of logic 150 years ago, but the basic ideas are the same. That is why propositional logic is also often called **Boolean logic** in recognition of Boole's ground-breaking contribution. It is not only a fundamental system of formal logical reasoning, but it also provides the mathematical basis of the **logical circuits** underlying the architecture of modern digital computers.

William Stanley Jevons (1.09.1835–13.08.1882) was an English economist and logician known for his pioneering works on political and mathematical economics, including the *theory of utility*. As well as contributing to the early development of modern logic, in 1869 he designed one of the first mechanical computers which he called the *logic piano*.

Jevons studied natural sciences and moral philosophy at the University College of London and, in 1866, was appointed Professor of Logic, Mental and Moral Philosophy and Professor of Political Economy at Owens College. His book *A General Mathematical Theory of Political Economy* (1862) is one of the first works on mathematical methods in economics which, being concerned with quantities, he regarded as an essentially mathematical science.

Jevons' most important work on scientific methods is his *Principles of Science* (1874). In 1870 he published *Elementary Lessons on Logic*, which soon became the most widely read elementary textbook on logic in the English language, later supplemented by his *Studies in Deductive Logic*.

Jevons developed a general theory of induction, which he regarded as an inverse method of deduction; he also developed and published his own treatments of Boole's approach to logic and on the general theory of probability, and studied the relation between probability and induction.

1.3 Logical equivalence: negation normal form of propositional formulae

1.3.1 Logically equivalent propositional formulae

Definition 16 *The propositional formulae A and B are* ***logically equivalent****, denoted $A \equiv B$, if for every assignment of truth values to the variables occurring in them they obtain the same truth values.*

Being a little imprecise (you'll see why), we can say that A and B are logically equivalent if they have the same truth tables.

- Every tautology is equivalent to $\top$. For example, $p \vee \neg p \equiv \top$.
- Every contradiction is equivalent to $\bot$. For example, $p \wedge \neg p \equiv \bot$.
- $\neg\neg p \equiv p$: a double negation of a proposition is equivalent to the proposition itself.
- $\neg(p \wedge q) \equiv (\neg p \vee \neg q)$ and $\neg(p \vee q) \equiv (\neg p \wedge \neg q)$. These are known as **De Morgan's laws**. Let us check the first, using simplified truth tables:

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

- $(p \wedge (p \vee q)) \equiv (p \wedge p)$. There is a small problem here: formally, the truth tables of these formulae are *not* the same, as the first contains two variables (p and q) while the second contains only p. However, we can always consider that q occurs *vacuously* in the second formula and include it in its truth table:

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

Checking logical equivalence can be streamlined, just like checking logical validity and consequence, by systematic search for a falsifying assignment, as follows. In order to check if $A \equiv B$ we try to construct a truth assignment to the variables occurring in A and B which renders one of them true while the other false. If such an assignment exists, the formulae are *not* logically equivalent; otherwise, they are. For example, let us check the second De Morgan's law: $\neg(p \vee q) \equiv (\neg p \wedge \neg q)$. There are two possibilities for an assignment to falsify that equivalence:

(i) $\neg(p \vee q) : \mathtt{T}$ and $(\neg p \wedge \neg q) : \mathtt{F}$. Then $p \vee q : \mathtt{F}$ hence $p : \mathtt{F}$, and $q : \mathtt{F}$, but then $(\neg p \wedge \neg q) : \mathtt{T}$: a contradiction.

(ii) $\neg(p \vee q) : \texttt{F}$, and $(\neg p \wedge \neg q) : \texttt{T}$. Then $\neg p : \texttt{T}$ and $\neg q : \texttt{T}$, hence $p : \texttt{F}$, and $q : \texttt{F}$. But then $\neg(p \vee q) : \texttt{T}$: a contradiction again.

There is therefore no falsifying assignment and the equivalence holds.

1.3.2 Basic properties of logical equivalence

Caution: do not confuse the propositional connective $\leftrightarrow$ and logical equivalence between formulae. These are different things: the former is a logical connective, a symbol in our *object language*, whereas the latter is a relation between formulae, that is, a statement in our *metalanguage*. However, as I show below, there is a simple relation between them.

1. Logical equivalence is reducible to logical validity:

$$A \equiv B \quad \text{iff} \quad \models A \leftrightarrow B.$$

   Indeed, they mean the same: that A and B always take the same truth values.
2. Logical equivalence is likewise reducible to logical consequence:

$$A \equiv B \quad \text{iff} \quad A \models B \quad \text{and} \quad B \models A$$

3. The relation $\equiv$ is an **equivalence relation**, that is, for every formulae A, B, C it is:
   (a) **reflexive**: $A \equiv A$;
   (b) **symmetric**: if $A \equiv B$ then $B \equiv A$;
   (c) **transitive**: if $A \equiv B$ and $B \equiv C$ then $A \equiv C$.
4. Moreover, $\equiv$ is a **congruence** with respect to the propositional connectives, that is:
   (a) if $A \equiv B$ then $\neg A \equiv \neg B$;
   (b) if $A_1 \equiv B_1$ and $A_2 \equiv B_2$ then $(A_1 \bullet A_2) \equiv (B_1 \bullet B_2)$, for $\bullet \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.
5. The following property of **equivalent replacement** holds. For any propositional formulae A, B, C and a propositional variable p, presumably occurring in C, if $A \equiv B$ then $C(A/p) \equiv C(B/p)$, where $C(X/p)$ is the result of simultaneous substitution of all occurrences of p by X.

Logical equivalence between propositional formulae can therefore be treated just like equality between algebraic expressions.

1.3.3 Some important logical equivalences

I now briefly present and discuss some important and useful logical equivalences. Verifying all of these are easy, but useful, exercises.

1.3.3.1 Algebraic laws for the logical connectives

I begin with some important logical equivalences which are used, for example, for equivalent transformations of propositional formulae to so-called *conjunctive and disjunctive normal forms* that I introduce later.

- **Idempotency**: $p \wedge p \equiv p$; $p \vee p \equiv p$.
- **Commutativity**: $p \wedge q \equiv q \wedge p$; $p \vee q \equiv q \vee p$.
- **Associativity**: $(p \wedge (q \wedge r)) \equiv ((p \wedge q) \wedge r)$; $(p \vee (q \vee r)) \equiv ((p \vee q) \vee r)$.
  This law allows parentheses to be omitted in multiple conjunctions and disjunctions.
- **Absorption**: $p \wedge (p \vee q) \equiv p$; $p \vee (p \wedge q) \equiv p$.
- **Distributivity**: $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$; $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$.

1.3.3.2 Equivalences mutually expressing logical connectives

The following logical equivalences, the proofs of which are left as easy exercises, can be used to define logical connectives in terms of others:

- $\neg A \equiv A \to \bot$. We sometimes use this equivalence in colloquial expressions, such as "If this is true then I can fly", when we mean "This cannot be true."
- $A \leftrightarrow B \equiv (A \to B) \wedge (B \to A)$. This equivalence allows us to consider the biconditional as definable connective, which we will often do.
- $A \vee B \equiv \neg(\neg A \wedge \neg B)$.
- $A \wedge B \equiv \neg(\neg A \vee \neg B)$.
- $A \to B \equiv \neg A \vee B$.
- $A \to B \equiv \neg(A \wedge \neg B)$.
- $A \vee B \equiv \neg A \to B$.
- $A \wedge B \equiv \neg(A \to \neg B)$.

We therefore see that each of $\wedge$, $\vee$, and $\to$ can be expressed by means of any other of these using negation.

1.3.3.3 Some simplifying equivalences

Other useful logical equivalences can be used to simplify formulae:

- $A \vee \neg A \equiv \top$, $A \wedge \neg A \equiv \bot$
- $A \wedge \top \equiv A$, $A \wedge \bot \equiv \bot$
- $A \vee \top \equiv \top$, $A \vee \bot \equiv A$
- $A \to \top \equiv \top$, $A \to \bot \equiv \neg A$
- $\top \to A \equiv A$, $\bot \to A \equiv \top$.
- $\neg A \to \neg B \equiv B \to A$ (Every implication is equivalent to its contrapositive.)

1.3.3.4 Negating propositional formulae: negation normal form

In mathematical and other arguments we sometimes have to negate a formalized statement and then use the result in further reasoning. For that, it is useful to transform, up to logical equivalence, the formula formalizing the statement in **negation normal form**, where negation may only occur in front of propositional variables. Such transformation can be done by step-by-step importing of all occurrences of negations inside the other logical connectives using the following equivalences, some of which we already know:

- $\neg\neg A \equiv A$
- $\neg(A \wedge B) \equiv \neg A \vee \neg B$
- $\neg(A \vee B) \equiv \neg A \wedge \neg B$
- $\neg(A \rightarrow B) \equiv A \wedge \neg B$
- $\neg(A \leftrightarrow B) \equiv (A \wedge \neg B) \vee (B \wedge \neg A)$.

Example 17 *Equivalent transformation to negation normal form:*

$\neg((A \vee \neg B) \rightarrow (\neg C \wedge D))$
$\equiv (A \vee \neg B) \wedge \neg(\neg C \wedge D)$
$\equiv (A \vee \neg B) \wedge (\neg\neg C \vee \neg D)$
$\equiv (A \vee \neg B) \wedge (C \vee \neg D)$.

References for further reading

To read more on propositional equivalence and negation normal form of propositional formulae see Nerode and Shore (1993), Jeffrey (1994), Barwise and Echemendy (1999), Hedman (2004), Nederpelt and Kamareddine (2004), Boole (2005), Chiswell and Hodges (2007), and Ben-Ari (2012).

Exercises

1.3.1 Verify the following logical laws:
(a) *Idempotency*: $p \wedge p \equiv p$, $p \vee p \equiv p$
(b) *Commutativity*: $p \wedge q \equiv q \wedge p$, $p \vee q \equiv q \vee p$
(c) *Associativity*: $(p \wedge (q \wedge r)) \equiv ((p \wedge q) \wedge r)$, $(p \vee (q \vee r)) \equiv ((p \vee q) \vee r)$
(d) *Absorption*: $p \wedge (p \vee q) \equiv p$, $p \vee (p \wedge q) \equiv p$
(e) *Distributivity*: $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$, $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
(f) *De Morgan's laws* $\neg(p \wedge q) \equiv (\neg p \vee \neg q)$, $\neg(p \vee q) \equiv (\neg p \wedge \neg q)$

1.3.2 Prove the following logical equivalences:

(a) $\neg(p \leftrightarrow q) \equiv (p \wedge \neg q) \vee (q \wedge \neg p)$
(b) $(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$
(c) $(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$
(d) $\neg(p \leftrightarrow q) \equiv (p \wedge \neg q) \vee (q \wedge \neg p)$
(e) $\neg(p \leftrightarrow q) \equiv \neg p \leftrightarrow q \equiv p \leftrightarrow \neg q$
(f) $p \rightarrow (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$
(g) $p \rightarrow (q \rightarrow r) \equiv q \rightarrow (p \rightarrow r)$
(h) $(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$
(i) $p \leftrightarrow q \equiv q \leftrightarrow p$
(j) $p \leftrightarrow (q \leftrightarrow r) \equiv (p \leftrightarrow q) \leftrightarrow r$

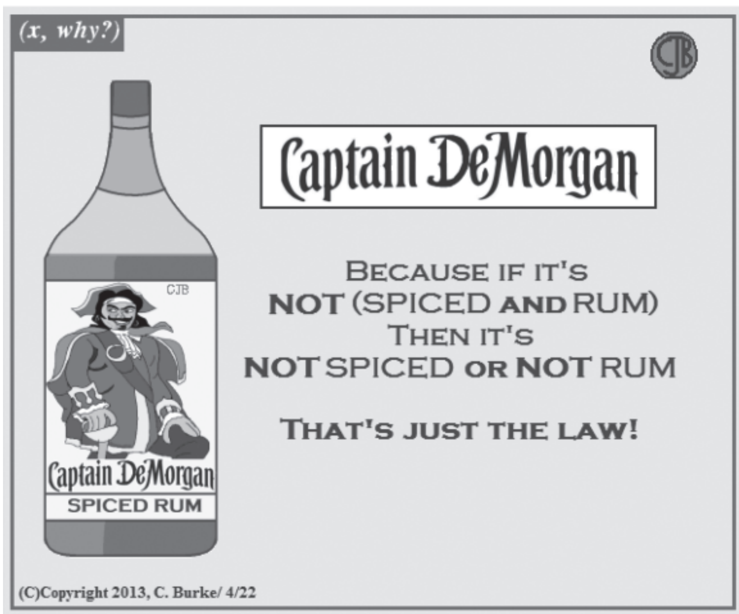
1.3.3 Determine which of the following pairs of formulae are logically equivalent:

(a) $p \rightarrow q$ and $\neg p \vee q$
(b) $\neg(p \rightarrow q)$ and $p \wedge \neg q$
(c) $\neg p \rightarrow \neg q$ and $q \rightarrow p$
(d) $p \rightarrow \neg q$ and $q \rightarrow \neg p$
(e) $\neg(p \rightarrow \neg q)$ and $p \wedge q$
(f) $((p \rightarrow q) \rightarrow q) \rightarrow q$ and $p \vee q$

(g) $(p \to r) \wedge (q \to r)$ and $(p \wedge q) \to r$
(h) $(p \to r) \vee (q \to r)$ and $(p \vee q) \to r$
(i) $((p \wedge q) \to r)$ and $(p \to r) \vee (q \to r)$
(j) $p \to (q \to r)$ and $(p \to q) \to r$
(k) $p \leftrightarrow (q \leftrightarrow r)$ and $q \leftrightarrow (p \leftrightarrow r)$
(l) $p \to (q \to r)$ and $(p \to q) \to (p \to r)$

1.3.4 Negate each of the following propositional formulae and transform the result to an equivalent formula in a negation normal form.

(a) $(p \vee \neg q) \wedge \neg p$
(b) $(p \to q) \vee (\neg p \to \neg q)$
(c) $(p \to \neg q) \to p$
(d) $p \to (\neg q \to p)$
(e) $(p \leftrightarrow \neg q) \to \neg r$
(f) $p \to (\neg q \leftrightarrow r)$



Augustus De Morgan (27.6.1806–18.3.1871) was a British mathematician, logician and a popularizer of mathematics. Influenced by George Boole, he pioneered the application of algebraic methods to the study of logic in the mid-19th century, becoming one of the founding fathers of modern mathematical logic. In particular, he was the first to formulate the logical equivalences now known as **de Morgan's laws**.

De Morgan was born in Madura, India and became blind in one eye soon after his birth. He graduated from Trinity

College, Cambridge and in 1828 became the first Professor of Mathematics at the newly established University College of London, where he taught for most of his academic life.

He was an enthusiastic and prolific writer of over 700 popular articles in mathematics for the *Penny Cyclopedia*, aiming to promote the education of mathematics in Britain. In an 1838 publication he formally introduced the term "**mathematical induction**" and developed the so-far informally used method of mathematical induction into a precise mathematical technique. He also wrote the books *Trigonometry and Double Algebra* and *The Differential and Integral Calculus*. In 1847 he published his main work on mathematical logic, *Formal Logic: The Calculus of Inference, Necessary and Probable*, which was used for a very long time and was last reprinted in 2003. De Morgan was also a passionate collector of mathematical puzzles, curiosities, and paradoxes, many of which he included in his book *A Budget of Paradoxes* published in 1872 (now digitalized and available on the internet).

In 1866 De Morgan became one of the founders and the first president of the London Mathematical Society. There is a crater on the Moon named after him.

Hugh MacColl (1831–1909) was a Scottish mathematician, logician, and novelist who made some important early contributions to modern logic.

MacColl grew up in a poor family in the Scottish Highlands and never obtained a university education because he could not afford it and refused to accept to take orders in the Church of England, a condition under which William Gladstone was prepared to support his education at Oxford. Consequently, he never obtained a regular academic position; he was a highly intelligent person however.

During 1877–1879 MacColl published a four-part article establishing the first-known variant of the propositional calculus, which he called the "calculus of equivalent statements", preceding Gottlob Frege's Begriffschrifft. Furthermore, MacColl's work on the nature of implication was later credited by C.I. Lewis as the initial inspiration of his own innovative work in modal logic. MacColl also promoted logical pluralism by exploring on a par ideas for several different logical systems such as modal logic, logic of fiction, connexive logic, many-valued logic, and probability logic, establishing himself as a pioneer in the field known as **non-classical logics** today.

1.4 Supplementary: Inductive definitions and structural induction and recursion

In section 1.1 I defined propositional formulae using a special kind of definition, which refers to the very notion it is defining. Such definitions are called *inductive*. They are very common and important, especially in logic, because they are simple, elegant, and indispensable when an infinite set of structured objects is to be defined. Moreover, properties of an object defined by inductive definitions can be proved by a uniform method, called *structural induction*, that resembles and extends the method of mathematical induction used to prove properties of natural numbers. Here I present the basics of the general theory of inductive definitions and structural induction. Part of this section, or even all of it, can be skipped, but the reader is recommended to read it through.

1.4.1 Inductive definitions

Let us begin with well-known cases: the inductive definition of words in an alphabet and then natural numbers as special words in a two-letter alphabet. Note the pattern.

1.4.1.1 The set of all finite words in an alphabet

Consider a set $\mathcal{A}$. Intuitively, a (finite) word in $\mathcal{A}$ is any string of elements of $\mathcal{A}$. We formally define the set of (finite) words in the alphabet $\mathcal{A}$ inductively as follows.

1. The empty string ϵ is a word in $\mathcal{A}$.
2. If w is a word in $\mathcal{A}$ and $a \in \mathcal{A}$, then wa is word in $\mathcal{A}$.

The idea of this definition is that words in $\mathcal{A}$ are those, and only those, objects that can be constructed following the two rules above.

1.4.1.2 The set of natural numbers

We consider the two-letter alphabet $\{0, S\}$, where $0, S$ are different symbols, and formally define natural numbers to be special words in that alphabet, as follows.

1. 0 is a natural number.
2. If n is a natural number then Sn is a natural number.

The definition above defines the infinite set $\{0, S0, SS0, SSS0, \cdots\}$.

Hereafter we denote $S \cdots_{n \text{ times}} \cdots S0$ by $\mathbf{n}$ and identify it with the (intuitive notion of) natural number n.

1.4.1.3 The set of propositional formulae

Let us denote the alphabet of symbols used in propositional logic $\mathcal{A}_{PL}$. Note that it includes a possibly infinite set PVAR of propositional variables.

We now revisit the inductive definition of propositional formulae as special words in the alphabet of symbols used in propositional logic, by paying closer attention to the structure of the definition. I emphasize the words *is a propositional formula* so we can see shortly how the definition transforms into an explicit definition and an induction principle.

Definition 18 *The property of a word in $\mathcal{A}_{PL}$ of being a propositional formula is defined inductively as follows.*

1. *Every Boolean constant (i.e., $\top$ or $\bot$) is a propositional formula.*
2. *Every propositional variable is a propositional formula.*
3. *If (the word) A is a propositional formula then (the word) $\neg A$ is a propositional formula.*
4. *If each of (the words) A and B is a propositional formula then each of (the words) $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, and $(A \leftrightarrow B)$ is a propositional formula.*

The meaning of the inductive definition above can be expressed equivalently by the following explicit definition, which essentially repeats the definition above but replaces the phrase "*is a propositional formula*" with "*is in (the set)* FOR."

Definition 19 *The set of propositional formulae FOR is the* least *set of words in the alphabet of propositional logic such that the following holds.*

1. *Every Boolean constant is in FOR.*
2. *Every propositional variable is in FOR.*
3. *If A is in FOR then $\neg A$ is in FOR.*
4. *If each of A and B is in FOR then each of $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, and $(A \leftrightarrow B)$ is in FOR.*

This pattern of converting the inductive definition into an explicit definition is general and can be applied to each of the other inductive definitions presented here. However, we have not yet proved that the definition of the set FOR given above is correct in the sense that the least (by inclusion) set described above even exists. Yet, if it does exist, then it is clearly unique because of being the least set with the described properties. We will prove the correctness later.

1.4.1.4 The subgroup of a given group, generated by a set of elements

I now provide a more algebraic example. The reader not familiar with the notions of groups and generated subgroups can skip this safely.

Let $\mathbf{G} = \langle G, \circ, ^{-1}, e \rangle$ be a group and X be a subset of G. The **subgroup of G generated by** X is the least subset $[X]_{\mathbf{G}}$ of G such that:

1. e is in $[X]_{\mathbf{G}}$.
2. Every element from X is in $[X]_{\mathbf{G}}$.

3. If $a \in [X]_{\mathbf{G}}$ then $a^{-1} \in [X]_{\mathbf{G}}$.
4. If $a, b \in [X]_{\mathbf{G}}$ then $a \circ b \in [X]_{\mathbf{G}}$.

Exercise: re-state the definition above as an inductive definition.

1.4.2 Induction principles and proofs by induction

With every inductive definition, a scheme for **proofs by induction** can be associated. The construction of this scheme is uniform from the inductive definition, as illustrated in the following.

1.4.2.1 Induction on the words in an alphabet

We begin with a principle of induction that allows us to prove properties of all words in a given alphabet. Given an alphabet $\mathcal{A}$, let $\mathcal{P}$ be a property of words in $\mathcal{A}$ such that:

1. The empty string ϵ has the property $\mathcal{P}$.
2. If the word w in $\mathcal{A}$ has the property $\mathcal{P}$ and $a \in \mathcal{A}$, then the word wa has the property $\mathcal{P}$.

Then, every word w in $\mathcal{A}$ has the property $\mathcal{P}$.

1.4.2.2 Induction on natural numbers

We can now formulate the well-known principle of mathematical induction on natural numbers in terms of the formal definition of natural numbers given above.

Let $\mathcal{P}$ be a property of natural numbers such that:

1. 0 has the property $\mathcal{P}$.
2. For every natural number n, if n has the property $\mathcal{P}$ then Sn has the property $\mathcal{P}$.

Then every natural number n has the property $\mathcal{P}$.

Here is the same principle, stated in set-theoretic terms:

Let $\mathcal{P}$ be a set of natural numbers such that:

1. $0 \in \mathcal{P}$.
2. For every natural number n, if $n \in \mathcal{P}$ then $Sn \in \mathcal{P}$.

Then every natural number n is in $\mathcal{P}$, that is, $\mathcal{P} = \mathbb{N}$.

1.4.2.3 Structural induction on propositional formulae

Following the same pattern, we can now state a principle of induction that allows us to prove properties of propositional formulae. Note that this principle is obtained almost automatically from the inductive definition of propositional formulae by replacing throughout that definition the words *is a propositional formula* with *satisfies the property* $\mathcal{P}$. Let $\mathcal{P}$ be a property of propositional formulae such that:

1. Every Boolean constant *satisfies the property* $\mathcal{P}$.
2. Every propositional variable *satisfies the property* $\mathcal{P}$.
3. If A *satisfies the property* $\mathcal{P}$ then $\neg A$ *satisfies the property* $\mathcal{P}$.
4. If each of A and B *satisfy the property* $\mathcal{P}$ then each of $(A \wedge B)$, $(A \vee B)$, $(A \to B)$, and $(A \leftrightarrow B)$ *satisfy the property* $\mathcal{P}$.

Then every propositional formula *satisfies the property* $\mathcal{P}$.

Again, the same principle can be formulated in set-theoretic terms by treating the property $\mathcal{P}$ as the set of those propositional formulae that satisfy it, and then replacing the phrase *satisfies the property* $\mathcal{P}$ with *is in the set* $\mathcal{P}$.

The induction principle can likewise be formulated for the elements of a subgroup of a given group, generated by a given set of elements. I leave that as an exercise.

1.4.3 Basics of the general theory of inductive definitions and principles

1.4.3.1 An abstract framework for inductive definitions

We extract the common pattern in the examples above to formulate a uniform abstract framework for inductive definitions and proofs by induction. The necessary ingredients for an inductive definition are:

- A **universe** U.
  In our examples, universes were sets of words in a given alphabet and the set of all elements of a given group.
- A subset $B \subseteq U$ of **initial (basic) elements**.
  In our examples, the sets of initial elements were: $\{\epsilon\}$; $\{0\}$; $\{\top, \bot\} \cup \mathrm{PVAR}$; and the set X of generators of a subgroup.
- A set **F** of **operations (constructors)** in U.
  In our examples, these were: the operation of appending a symbol to a word; the operation of prefixing S to a natural number; the propositional logical connectives regarded as operations on words; and the group operations.

We fix the sets $U, B, \mathbf{F}$ arbitrarily thereafter. Our aim is to define formally the set of elements of U inductively defined over B by applying the operations in F, denoted $\mathcal{C}(B, \mathbf{F})$. Intuitively, this will be the set defined by the following inductive definition:

1. Every element of B is in $\mathcal{C}(B, \mathbf{F})$.
2. For every operation $f \in \mathbf{F}$ such that $f : U^n \to U$, if every $x_1, \ldots, x_n$ is in $\mathcal{C}(B, \mathbf{F})$ then $f(x_1, \ldots, x_n)$ is in $\mathcal{C}(B, \mathbf{F})$.

We give the set $\mathcal{C}(B, \mathbf{F})$ a precise mathematical meaning by defining it in two different, yet eventually equivalent, ways.

1.4.3.2 Top-down closure construction

Definition 20 *A set $C \subseteq U$ is:*

1. ***closed under the operation*** $f \in \mathbf{F}$*, such that* $f : U^n \to U$*, if* $f(x_1, \dots, x_n) \in C$ *for every* $x_1, \dots, x_n \in C$.
2. ***closed****, if it is closed under every operation* $f \in \mathbf{F}$.
3. ***inductive****, if* $B \subseteq C$ *and* C *is closed.*

Remark 21 *The elements of B can be regarded as constant (0-argument) functions in U, and the condition $B \subseteq C$ can therefore be subsumed by closedness.*

Proposition 22 *Intersection of any family of inductive sets is an inductive set.*

Proof. I leave this as an exercise.

Definition 23 *C^* is the intersection of the family of all inductive sets.*

By Proposition 22, C^* is the smallest inductive set.

1.4.3.3 Bottom-up inductive construction

Definition 24 *A **construction tree** for an element $x \in U$ is a finite tree $T(x)$, every node of which is labeled with an element of U and the successors of every node are ordered linearly, satisfying the following conditions:*

1. *Every leaf in $T(x)$ is labeled by an element of B.*
2. *If a node in $T(x)$ labeled with y has k successors labeled by elements listed in the order of successors $y_1, \dots, y_k$, then there is a k-ary operation $f \in \mathbf{F}$ such that $y = f(y_1, \dots, y_k)$.*
3. *The root of $T(x)$ is labeled by x.*

Definition 25 *The **height** of a finite tree is the length (number of nodes minus 1) of the longest path in the tree.*

Definition 26 *The **rank of an element** $x \in U$ is the least height $\mathbf{r}(x)$ of a construction tree for x if it exists; otherwise, the rank is ∞.*

Definition 27 *We define a hierarchy of sets $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_*$ as follows.*

1. *C_n is the set of all elements of U with rank $\leq n$.*
2. $C_* := \bigcup_{n \in \mathbf{N}} C_n$.

Proposition 28 *The following holds.*

1. $C_0 = B$.
2. $C_{n+1} = C_n \cup \{f(x_1, \ldots, x_n) \mid f \in \mathbf{F}$ *is an* n*-ary operation on* U *and* $x_1, \ldots, x_n \in C_n\}$

Proof. Easy induction on the rank n, which I leave as an exercise for the reader. ■

1.4.3.4 Inductive definitions

Proposition 29 C_* *is an inductive set.*

Proof. Exercise. ■

Corollary 30 $C^* \subseteq C_*$*, since* C^* *is the least inductive set.*

Proposition 31 $C_* \subseteq C^*$.

Proof. We prove by induction on n that $C_n \subseteq C^*$. Exercise. ■

Definition 32 $C^*(= C_*)$ *is the* ***set inductively defined over*** B ***and*** F*, denoted* $\mathcal{C}(B, \mathbf{F})$.

1.4.3.5 Induction principle for inductively defined sets

We can easily generalize the ordinary principle of mathematical induction on natural numbers to induction in the set $\mathcal{C}(B, \mathbf{F})$.

Proposition 33 (Induction principle for $\mathcal{C}(B, \mathbf{F})$) *Let* $\mathcal{P}$ *be a property of elements of* U*, such that:*

1. *Every element of* B *has the property* $\mathcal{P}$.
2. *For every operation* $f \in \mathbf{F}$*, such that* $f : U^n \rightarrow U$*, if every* $x_1, \ldots, x_n$ *has the property* $\mathcal{P}$ *then* $f(x_1, \ldots, x_n)$ *has the property* $\mathcal{P}$.

Then every element of $\mathcal{C}(B, \mathbf{F})$ *has the property* $\mathcal{P}$.

The proof is straightforward. The two conditions above state precisely that the set of elements of $\mathcal{C}(B, \mathbf{F})$ that have the property $\mathcal{P}$ is inductive; it therefore contains the least inductive set, that is, $\mathcal{C}(B, \mathbf{F})$.

It is quite easy to see that the induction principle above generalizes all those formulated earlier for our examples.

1.4.4 Inductive definitions and proofs in well-founded sets

Here I briefly generalize the induction principle from inductively defined sets to the more abstract notion of *well-founded sets*.

1.4.4.1 Well-founded sets

Definition 34 *A partially ordered set (poset)* $(X, <)$ *is* ***well-founded*** *if it contains no infinite strictly descending sequences* $x_1 > x_2 > \ldots$.

A well-founded linear ordering is called a ***well-ordering****.*

Example 35

- *Every finite poset is well-founded.*
- $\langle \mathbf{N}, < \rangle$ *is well-ordered, while* $\langle \mathbf{Z}, < \rangle$ *and* $\langle \mathbf{Q}, < \rangle$ *are not.*
- *The poset* $(\boldsymbol{P}(X), \subseteq)$*, where* X *is any infinite set, is not well-founded.*
- *The* ***lexicographic ordering*** *in* $\mathbf{N}^2$ *defined by* $\langle x_1, y_1 \rangle \leq \langle x_2, y_2 \rangle$ *iff* $x_1 < x_2$ *or* ($x_1 = x_2$ *and* $y_1 \leq y_2$) *is a well-ordering in* $\mathbf{N}^2$.
- *The relation "*A *is a strict subformula of* B*" in the set* FOR *is a well-ordering.*

Proposition 36 *A poset* $\langle X, < \rangle$ *is well-founded iff every non-empty subset of* X *has a minimal element. Respectively, a linear ordering* $\langle X, < \rangle$ *is a well-ordering iff every non-empty subset of* X *has a least element.*

1.4.4.2 Induction principle for well-founded sets

Let $(X, <)$ be a well-founded poset. The **induction principle** for $(X, <)$ states the following:

Let $\mathcal{P} \subseteq X$ be such that for every $x \in X$, if all elements of X less than x belong to $\mathcal{P}$ then x itself belongs to $\mathcal{P}$. Then $\mathcal{P} = X$.

Proof. Assume the contrary, that is, $X - \mathcal{P} \neq \phi$. Then $X - \mathcal{P}$ has a minimal element x. Then all elements of X less than x belong to $\mathcal{P}$, hence x must belong to $\mathcal{P}$: a contradiction. ■

Example 37 *Let* $\mathcal{P}$ *be a property (set) of propositional formulae such that for every formula* $A \in FOR$*, if all strict subformulae of* X *have the property (belong to the set)* $\mathcal{P}$*, then* A *itself has the property (belongs to the set)* $\mathcal{P}$*.*

Then every formula $A \in FOR$ *has the property (belongs to the set)* $\mathcal{P}$*.*

1.4.5 Recursive definitions on inductively definable sets

We now consider the following general problem: given an inductively defined set $\mathcal{C}(B, \mathbf{F})$, how should we define a function h on that set by using the inductive definition? The idea is to first define h on the set B, and then provide rules prescribing how the definition of that function propagates over all operations. Formally, in order to define by recursion a mapping $h : \mathcal{C}(B, \mathcal{F}) \to X$ where X is a fixed target set, we need:

1. A mapping $h_0 : B \to X$.
2. For every n-ary operation $f \in \mathcal{F}$ a mapping $F_f : X^{2n} \to X$.

We now define the mapping h as follows:

1. If $a \in B$ then $h(a) := h_0(a)$.
2. For every n-ary operation $f \in \mathcal{F}$:

$$h(f(a_1, \ldots, a_n)) := F_f(a_1, \ldots, a_n, h(a_1), \ldots, h(a_n)).$$

We will soon discuss the meaning and correctness of such definitions, but let us first look at some important particular cases.

1.4.5.1 Primitive recursion on natural numbers

Functions on natural numbers can be defined by so-called *primitive recursion* using the inductive definition provided earlier in this section.

1. The basic scheme of primitive recursion is:

$$h(0) = a$$
$$h(n+1) = h(Sn) = F_S(n, h(n)).$$

For example, the scheme

$$h(0) = 1$$
$$h(Sn) = (n+1)h(n)$$

defines the factorial function $h(n) = n!$.

2. The more general scheme of primitive recursion with parameters is:

$$h(\mathbf{m}, 0) = F_0(\mathbf{m}).$$
$$h(\mathbf{m}, Sn) = F_S(\mathbf{m}, n, h(\mathbf{m}, n)).$$

For example, the scheme

$$h(m, 0) = m,$$
$$h(m, n+1) = h(m, n) + 1$$

defines the function addition $h(m, n) = m + n$.

1.4.5.2 Truth valuations of propositional formulae

Recall that the set of propositional formulae FOR is built on a set of propositional variables PVAR and a truth assignment is a mapping $s : \text{PVAR} \to \{\mathtt{T}, \mathtt{F}\}$. Now, given any truth assignment $s : \text{PVAR} \to \{\mathtt{T}, \mathtt{F}\}$ we can define a mapping $\alpha : \text{FOR} \to \{\mathtt{T}, \mathtt{F}\}$ that extends it to a **truth valuation**, a function computing the truth values of all formulae in FOR by recursion on the inductive definition of FOR as follows:

1. $\alpha(\mathbf{t}) = \mathtt{T}$, $\alpha(\mathbf{f}) = \mathtt{F}$.
2. $\alpha(p) = s(p)$ for every propositional variable p.
3. $\alpha(\neg A) = F_\neg(\alpha(A))$,
   where $F_\neg : \{\mathtt{T}, \mathtt{F}\} \to \{\mathtt{T}, \mathtt{F}\}$ is defined as follows: $F_\neg(\mathtt{T}) = \mathtt{F}$, $F_\neg(\mathtt{F}) = \mathtt{T}$.
4. $\alpha(A \wedge B) = F_\wedge(\alpha(A), \alpha(B))$,
   where $F_\wedge : \{\mathtt{T}, \mathtt{F}\}^2 \to \{\mathtt{T}, \mathtt{F}\}$ is defined as follows:
   $F_\wedge(\mathtt{T}, \mathtt{T}) = \mathtt{T}$ and $F_\wedge(\mathtt{T}, \mathtt{F}) = F_\wedge(\mathtt{F}, \mathtt{T}) = F_\wedge(\mathtt{F}, \mathtt{F}) = \mathtt{F}$.
   (That is, $F_\wedge$ computes the truth table of $\wedge$.)
5. $\alpha(A \vee B) = F_\vee(\alpha(A), \alpha(B))$,
   where $F_\vee : \{\mathtt{T}, F\}^2 \to \{\mathtt{T}, \mathtt{F}\}$ is defined as follows:
   $F_\vee(\mathtt{T}, \mathtt{T}) = F_\vee(\mathtt{T}, \mathtt{F}) = F_\vee(\mathtt{F}, \mathtt{T}) = \mathtt{T}$ and $F_\vee(\mathtt{F}, \mathtt{F}) = \mathtt{F}$.
6. $\alpha(A \to B) = F_\to(\alpha(A), \alpha(B))$,
   where $F_\to : \{\mathtt{T}, F\}^2 \to \{\mathtt{T}, \mathtt{F}\}$ is defined according to the truth table of $\to$.
7. $\alpha(A \leftrightarrow B) = F_\leftrightarrow(\alpha(A), \alpha(B))$,
   where $F_\leftrightarrow : \{\mathtt{T}, F\}^2 \to \{\mathtt{T}, \mathtt{F}\}$ is defined according to the truth table of $\leftrightarrow$.

The mapping α so defined is called *the truth valuation* of the propositional formulae generated by the truth assignment s.

Using such recursive definitions, we can likewise define various other natural functions associated with propositional formulae such as length, number of occurrences of logical connectives, and set of occurring propositional variables. I leave these as exercises.

1.4.5.3 Homomorphisms on freely generated groups

Given a group **G** with a set of free generators B and any group **H**, every mapping $h_0 : B \to \mathbf{H}$ can be (uniquely) extended to a homomorphism $h : \mathbf{G} \to \mathbf{H}$. The definition of h is essentially by recursion on the inductive definition of **G** as generated by B, and I leave it as an exercise.

1.4.5.4 Other inductive definitions and recursion on natural numbers

Consider the following inductive definitions:

Definition 38

1. *0 is a natural number.*
2. *If n is a natural number then $2n + 1$ is a natural number.*
3. *If n is a natural number and $n > 0$ then $2n$ is a natural number.*

Definition 39

1. *0 is a natural number.*
2. *If n is a natural number then $n + 2$ is a natural number.*
3. *If n is a natural number then $2n + 1$ is a natural number.*

I leave it as exercises for the reader to show that each of these inductive definitions defines the set of all natural numbers.

We now consider the recursive definitions:

1. $h_1(0) = 1$;
2. $h_1(2n + 1) = 3n + h_1(n)$;
3. $h_1(2n) = h_1(n) + 1$, for $n > 0$.

and

1. $h_2(0) = 0$;
2. $h_2(n + 2) = 2h_2(n) + 3$;
3. $h_2(2n + 1) = h_2(n) + 1$;

They look similar, and yet there is something wrong with the second definition. What? First, note that $h_2(1) = h_2(2 \times 0 + 1) = h_2(0) + 1 = 1$. Now, let us compute $h_2(3)$. On the one hand, $h_2(3) = h_2(1 + 2) = 2h_2(1) + 3 = 5$. On the other hand, $h_2(3) = h_2(2 \cdot 1 + 1) = h_2(1) + 1 = 2$. Thus, we have obtained two different values, which is definitely bad. The problem comes from the fact that the second definition allows for essentially different generations of the same object, leading us to define the notion of **unique generation**.

1.4.5.5 Unique generation

For the correctness of recursive definitions it should be required that every element of $\mathcal{C}(B, \mathbf{F})$ can be constructed uniquely (up to the order of the steps). Otherwise, definitions can lead to problems as above. More formally, the elements of $\mathcal{C}(B, \mathbf{F})$ are represented by expressions (terms) built from the elements of B by applying the operations from $\mathbf{F}$. Unique generation means that every element of $\mathcal{C}(B, \mathbf{F})$ can be represented by a unique expression.

Example 40

1. *The standard definition of natural numbers and Definition 1 above have the unique generation property. These can be proved by induction on natural numbers.*
2. *Definition 2 given above does not satisfy the unique generation property.*
3. *The set FOR of propositional formulae satisfies the unique generation property, also known as* ***unique readability property****.*
4. *If the subgroup* ***H*** *of a group* ***G*** *is freely generated by a set of generators B, then it satisfies the unique generation property.*
5. *If the subgroup* ***H*** *of a group* ***G*** *is not freely generated by a set of generators B, then it does not satisfy the unique generation property.*

Theorem 41 *If $\mathcal{C}(B, \mathbf{F})$ satisfies the unique generation property then for every mapping $h_0 : B \to X$ and mappings $\{F_f : X^{2n} \to X \mid f \in \mathcal{F}\}$ there exists a unique mapping $h : \mathcal{C}(B, \mathcal{F}) \to X$ defined by the recursive scheme in Section 1.4.5.*

I do not give a proof here, but refer the reader to Enderton (2001).

References for further reading

For further reading in inductive definitions, structural induction, and recursion see Tarski (1965), Shoenfield (1967), Barwise and Echemendy (1999), Enderton (2001), Hedman (2004), Nederpelt and Kamareddine (2004), and Makinson (2008).

Exercises

1.4.1 Given a set Z, give an inductive definition of the set of *lists of elements of* Z. Formally, these are special words in the alphabet $Z \cup \{[\ ,\ \ ;\ \ ,\ ]\}$ of the type $[\cdots[]; z_1]; z_2]; \cdots z_n]$, where [] is the empty list and $z_1, z_2, \ldots, z_n \in Z$.
Then formulate the induction principle for the set of lists of elements of Z, both for properties of lists and for sets of lists, and use it to prove that every list of elements of Z has equal numbers of occurrences of [and of].

1.4.2 Rephrase the definition of the subgroup $[X]_{\mathbf{G}}$ of a group $\mathbf{G}$ generated by a set X, given in this section, as an inductive definition.
Then formulate the induction principle for the elements of $[X]_{\mathbf{G}}$, both for properties and for sets of elements of the group.

1.4.3 Prove by structural induction that every propositional formula has equal numbers of occurrences of (and).

1.4.4 Prove Proposition 22.

1.4.5 Prove Proposition 28.

1.4.6 Prove Proposition 31.

1.4.7 Use addition and primitive recursion to define multiplication on natural numbers.

1.4.8 Use multiplication and primitive recursion to define exponentiation on natural numbers.

1.4.9 Complete the definition of the mappings $F_{\rightarrow}$ and $F_{\leftrightarrow}$ in the recursive definition of truth valuations of propositional formulae.

1.4.10 Given a group $\mathbf{G}$ with a set of free generators B, a group $\mathbf{H}$, and a mapping $h_0 : B \rightarrow \mathbf{H}$, define the unique homomorphism $h : \mathbf{G} \rightarrow \mathbf{H}$ extending h_0 by recursion on the inductive definition of $\mathbf{G}$ as generated by B.

1.4.11 Show that each of the alternative inductive definitions of the set of natural numbers given in Section 1.4.5.4 is correct.

1.4.12 Prove by induction on natural numbers that each of the standard definitions of natural numbers and Definition 1 in Section 1.4.5.4 satisfy the unique generation property.

1.4.13 Prove by structural induction on FOR that the set FOR of propositional formulae satisfies the unique generation property.

1.4.14 Using the inductive definition of FOR, give recursive definitions of the following functions associated with propositional formulae:
(a) the length of a formula, being the number of symbols occurring in it;
(b) the number of occurrences of logical connectives in a formula;
(c) the set of propositional variables occurring in a formula;
(d) the set of logical connectives occurring in a formula; and
(e) the nesting depth of a formula, being the largest number of logical connectives occurring in the scope of one another in the formula.

Ludwig Wittgenstein (26.04.1889–29.04.1951) was one of the most prominent philosophers and logicians of the 20th century. His work spanned logic, philosophy of mathematics, philosophy of mind, and philosophy of language. During his lifetime he published just one book in philosophy, the extremely influential (and quite enigmatic) 75-page long *Tractatus Logico-Philosophicus* in 1921, but he produced and wrote much more, mostly published only in 1953 after his death, in the book *Philosophical Investigations*.

Wittgenstein was born in Vienna into one of Europe's richest families. He completed high school in Linz, Austria and went on to study mechanical engineering in Berlin and then aeronautics in Manchester. He gradually developed a strong interest in logic and the foundations of mathematics, influenced by Russell's *Principia Mathematica* and Frege's *Grundgesetze der Arithmetik*. In 1911 he visited Frege in Jena and wanted to study with him; they did not quite match each other however. He went on to Cambridge to study logic with Russell, who quickly recognized Wittgenstein as a genius. However, Wittgenstein became depressed in Cambridge and in 1913 went to work in isolation in Skjolden, Norway, where he conceived much of *Tractatus*. When World War I broke out, Wittgenstein returned to Vienna to join the army and fight in the war for 4 years. During that period he received numerous military decorations for his courage.

During the war, Wittgenstein completed the manuscript of his *Tractatus*, where he presented his philosophical theory on how logic, language, mind, and the real world relate. In a simplified summary, he argued that words are just representations of objects and propositions are just words combined to make statements (pictures) about reality, which may be true or false, while the real world is nothing but the facts that are in it. These facts can then be reduced to *states of affair* (later leading to the concept of "possible worlds"), which in turn can be reduced to combinations of objects, thus eventually creating a precise correspondence between language and the world. Wittgenstein believed he had solved all philosophical problems in his *Tractatus*. In any case, he had essentially invented propositional formulae and truth tables. In his later period, Wittgenstein deviated from some of his views in *Tractatus*. In particular, he came to believe that the meaning of words was entirely in their use.

Wittgenstein sent his *Tractatus* to Russell from Italy in 1918, where he was detained as a prisoner of war, but only submitted it as his doctoral thesis when he returned to Cambridge in 1929. During a period of depression after the war in 1919, he gave the fortune that he had inherited from his father to his brothers and sisters. He then he had to support himself by working at various jobs, including as maths teacher in a remote Austrian village, a gardener at a monastery, and an architect for his sister's house. He remained in Cambridge to teach during 1929–1947, which was his longest period in academia. Interrupted by World War II, he decided that it was not morally justifiable to stay in the comfortable academic world in such troubled times, worked anonymously as a hospital porter in London (where he advised patients not to take the drugs they were prescribed). He resigned from Cambridge in 1947 and retired in isolation to a remote cottage on the coast of Ireland, where he died of cancer in 1951. Despite his worsening health he worked until the very end, which he is said to have accepted quite willingly as he was apparently never quite happy with his life and felt he did not really fit into this world.

To the present day, Wittgenstein remains one of the most studied and discussed, but also probably one of the least understood, modern thinkers.