Logic as a Tool

A Guide to Formal Logical Reasoning

VALENTIN GORANKO

WILEY

LOGIC AS A TOOL

LOGIC AS A TOOL

A GUIDE TO FORMAL LOGICAL REASONING

Valentin Goranko
Stockholm University, Sweden

WILEY

This edition first published 2016

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

Library of Congress Cataloging-in-Publication Data

Names: Goranko, Valentin, author.
Title: Logic as a tool : a guide to formal logical reasoning / Valentin Goranko.
Description: Chichester, UK ; Hoboken, NJ : John Wiley & Sons, 2016. | Includes bibliographical references and index.
Identifiers: LCCN 2016010458 (print) | LCCN 2016014532 (ebook) | ISBN 9781118880005 (cloth) | ISBN 9781118880050 (pdf) | ISBN 9781118880043 (epub)
Subjects: LCSH: Logic–Textbooks.
Classification: LCC BC71 .G67 2016 (print) | LCC BC71 (ebook) | DDC 511.3–dc23
LC record available at http://lccn.loc.gov/2016010458

A catalogue record for this book is available from the British Library.

Set in 10.5/12pt, TimesLTStd by SPi Global, Chennai, India.

1 2016

This book is dedicated to those from whom I have learned
and to those who will learn from it.

Contents

Preface

Unlike most books and textbooks on logic, this one purports to teach logic not so much as a *subject to study*, but rather as a *tool to master and use* for performing and structuring correct reasoning. It introduces classical logic rather informally, with very few theorems and proofs (which are mainly located in the supplementary sections). Nevertheless, the exposition is systematic and precise, without compromising on the essential technical and conceptual issues and subtle points inherent in logic.

Aims

This textbook covers only the core of *classical logic*, which itself is just the heart of the vast and growing body of modern logic. The main aims of the book are:

1. to explain the language, grammar, meaning, and formal semantics of logical formulae, to help the reader understand the use of classical logical languages and be able both to formalize natural language statements in them and translate back from logical formulae to natural language;
2. to present, explain, and illustrate with examples the use of the most popular deductive systems (namely, axiomatic systems, Semantic Tableaux, Natural Deduction, and Resolution with the only notable exclusion being Sequent Calculus, which is essentially inter-reducible with Natural Deduction) for mechanizing and "computing" logical reasoning both on propositional and on first-order level, and to provide the reader with the necessary technical skills for practical derivations in them; and
3. to offer systematic advice and guidelines on how to organize and perform a logically correct and well-structured reasoning using these deductive systems and the reasoning techniques that they provide.

Summary of the content and main features

The structure of the book reflects the two levels of expression and reasoning in classical logic: propositional and first-order.

The first two chapters are devoted to propositional logic. In Chapter 1 I explain how to understand propositions and compute their truth values. I then introduce propositional formulae and their truth tables and then discuss logical validity of propositional arguments. The fundamental notion here is that of *propositional logical consequence*. Then, in Chapter 2, I present several *deductive systems* used for deriving logical consequences in propositional logic and show how they can be used for checking the logical correctness of propositional arguments and reasoning. In a supplementary section at the end of the chapter I sketch generic proofs of soundness and completeness of the propositional deductive systems.

The exposition of propositional logic is uplifted to first-order logic in the following two chapters. In Chapter 3 I present first-order structures and languages and then the syntax and semantics (first informally, and then more rigorously) of first-order logic. Then I focus on using first-order languages and translations between them and natural languages. In the last section of this chapter I present and discuss the fundamental semantic concepts of logical validity, consequence, and equivalence in first-order logic. Deductive systems for first-order logic are introduced in Chapter 4 by extending the respective propositional deductive systems with additional rules for the quantifiers. Derivations in each of these are illustrated with several examples. Again in a supplementary section, I sketch generic proofs of soundness and completeness of the deductive systems for first-order logic.

Chapter 5 contains some applications of classical logic to mathematical reasoning and proofs, first in general and then specifically, for sets functions, relations, and arithmetic. It consists of concise presentations of the basic theories of these, where the proofs are left as exercises. The chapter ends with applications of classical logic to automated reasoning and theorem proving, as well as to logic programming, illustrated briefly with Prolog.

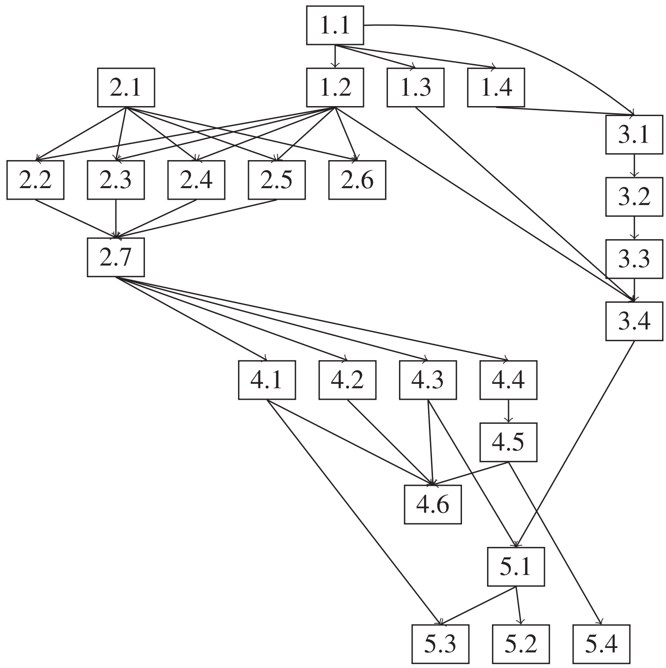
The book ends with a comprehensive set of detailed solutions or answers to many of the exercises.

The special features of this book include:

- concise exposition, with semi-formal but rigorous treatment of the minimum necessary theory;
- emphasis both on conceptual understanding by providing many examples, and on developing technical skills and building experience by providing numerous exercises, most of them standard, yet non-trivial, as well as full solutions or answers for many of them;
- solid and balanced coverage of semantic, syntactic, and deductive aspects of logic;
- some refreshing extras, such as a few logic-related cartoons scattered around, as well as many biographical boxes at the end of each section with photos and short texts on distinguished logicians, providing some background to their lives and contributions;
- selected references to other books on logic, listed at the end of each section, which are suitable for further reading on the topics covered in the section; and
- a supplementary website with slides, additional exercises, more solutions, and errata, which can be viewed at https://logicasatool.wordpress.com

For the instructor

The textbook is intended for introductory and intermediate courses in classical logic, mainly for students in both mathematics and computer science, but is also suitable and useful for more technically oriented courses for students in philosophy and social sciences.



Dependancy chart

Some parts of the text and some exercises are much more relevant to only one of the main target audiences, and I have indicated them by using *Mathematics Track* and *Computer Science Track* markers in the text. Everything else which is not explicitly included in either of these tracks should be suitable for both groups. Likewise, some specific topics and exercises are somewhat more advanced and are indicated with an *Advanced Track* marker. These are, of course, only indications.

The whole book can be covered in one or two semester courses, depending on the background and technical level of the audience. It assumes almost no specific prior knowledge, except some general background in college maths for specific topics and examples, usually indicated in *Mathematics* or *Advanced* tracks. A dependency chart of the different sections is provided in the figure above.

Acknowledgements

This textbook grew out of lecture notes that I have compiled for introductory logic courses for students in mathematics and computer science since the late 1990s. Many people have contributed in various ways to this book over these years. I am grateful to former colleagues who have helped with valuable comments, technical or editorial corrections, and some solutions to exercises, including Thomas Bolander, Willem Conradie, Ruaan Kellerman and Claudette Robinson, as well as to many students at the University of Johannesburg, the University of the Witwatersrand, and the Technical University of Denmark, who have sent me useful feedback and have noticed some errors in the lecture notes from which this book has evolved.

I also thank Christopher Burke, Mike Cavers, and Andreï Kostyrka for generously allowing me to include some of their cartoons in the book.

I gratefully acknowledge Wikipedia, the Stanford Encyclopedia of Philosophy, and the MacTutor History of Mathematics archive of the School of Mathematics and Statistics University of St Andrews as the main sources of the historical and biographical information provided.

The core content of the present book is a substantially extended version of the two chapters in logic in the recently published *Logic and Discrete Mathematics: A Concise Introduction* (written by Willem Conradie and myself, published by Wiley). I am grateful to Wiley for the continued collaboration. In particular, I thank everyone employed or contracted by Wiley who took part in the different stages of the technical preparation of this book.

Lastly, I owe very special thanks to my life partner Nina for her personal support and understanding during the work on this book, as well as for her invaluable technical help with producing many figures and diagrams and collecting photos and references for many of the historical boxes and the cover of the book.

Attributions for the photos used in the book

Most of the photos are public domain or under Creative Commons licence with free permissions. The photos of Cook, Fitting, Prawitz and Hodges are obtained from their owners or used with their permission. Attributions for the rest are listed below.

Davis
Author of photo: David Monniaux

Gentzen
Author of photo: "Eckart Menzler-Trott", permission obtained from the MFO "Archives of the Mathematisches Forschungsinstitut Oberwolfach"

Henkin
Author of photo: George M. Bergman

Herbrand
Author of photo: Journal des Sciences

Jaśkowski
Author of photo: Alojzy Czarnecki

Kleene
Author of photo: Konrad Jacobs, Erlangen, permission obtained from the MFO "Archives of the Mathematisches Forschungsinstitut Oberwolfach"

Levin
Author of photo: Wikipedia user Sergio01

Maltsev
Author of photo: Shiltsev

Robinson
Author of photo: David Monniaux

Putnam
Author of photo: Hilary Putnam

Skolem
Author of photo: Oslo Museum: image no. OB.F06426c (Byhistorisk samling/City historic collection), via oslobilder.no

Tarski
Author of photo: George M. Bergman, permission obtained from the MFO "Archives of the Mathematisches Forschungsinstitut Oberwolfach"

Wittgenstein
Author of photo: Moritz Nähr

Zeno
Author of photo: Wikipedia user Shakko

Introduction

What is logic about? What does it study and what does it offer? A usual definition found in the encyclopedia is that it is the branch of philosophy that studies the laws and rules of human reasoning. Little of this is actually correct. First, logic left the cradle of philosophy long ago and is now a truly interdisciplinary area, related and relevant not only to philosophy but also to mathematics, computer science, artificial intelligence, linguistics, social sciences, and even economics. Second, is logic really about how we reason? If that were the case, as a professional logician for many years I should already know quite well how exactly humans reason. Alas, the more experience I gain in life, the less I understand that. One thing is certain: most people use in their everyday reasoning emotions, analogies, clichés, ungrounded beliefs and superstitions, that is, everything but logic. But then, maybe logic studies the reasoning of the *rational* human, for whom reasoning is a purely rational brain activity? Well, while many (but far from all) of us humans reason with their brains, this is not sufficient to understand how we do it. As the American scientist Emerson M. Pugh brilliantly put it: "If the human brain were so simple that we could understand it, we would be so simple that we couldn't."

What does logic tell us, after all, if not how we reason? A better answer is: it tells us how we *can* – and ideally *should* – reason in a systematic and well-structured way that would *guarantee* that we always derive true and correct conclusions, providing we *only* use true assumptions and *only* apply logically correct rules of reasoning. Logic is therefore not just concerned with what is true and what is false, but rather with the *correctness* of our argumentation of what implies what and with the *validity* of our reasoning. What exactly does all that mean? This book aims to answer this question, beginning with some food for thought here.

The famous Greek philosopher Aristotle (384–322 BC), regarded as the founding father of formal logic, was the first who systematically studied and classified logically correct and incorrect forms and rules of reasoning. Aristotle studied specific patterns of arguments called *syllogisms*. Here is a typical example (mine, *not* Aristotle's) of a syllogism:

$$\frac{\begin{array}{l}\text{All logicians are clever.}\\ \text{All clever people are rich.}\end{array}}{\text{All logicians are rich.}}$$

The way we actually read this is as follows.

If all logicians are clever **and** all clever people are rich, **then** all logicians are rich.

This sounds intuitively like a correct piece of reasoning, and it *is*, but it does not mean that the conclusion is necessarily *true*. (In fact, unfortunately, it is not.) What then makes it correct?

Here is another example:

$$\frac{\begin{array}{l}\text{All natural numbers are integers.}\\ \text{Some integers are odd numbers.}\end{array}}{\text{Some natural numbers are odd numbers.}}$$

Note that all statements above are true. So, is this a correct argument? If you think so, then how about taking the same argument and replacing the words "natural numbers" by "mice," "integers" by "animals," and "odd numbers" by "elephants." This will not change the logical shape of the argument and, therefore, should not change its logical correctness. The result speaks for itself, however.

$$\frac{\begin{array}{l}\text{All mice are animals.}\\ \text{Some animals are elephants.}\end{array}}{\text{Some mice are elephants.}}$$

So what makes an argument logically correct? You will also find answers to this question in this book.

Let me say a few more concrete words about the main aspects and issues of classical logic treated in this book. There are two levels of logical discourse and reasoning in classical logic. The lower level is *propositional logic*, introduced and discussed in the first two chapters of this book, and the higher level is *first-order logic*, also known as *predicate logic*, treated in the rest of the book.

Propositional logic is about reasoning with *propositions*, sentences that can be assigned a *truth value* of either *true* or *false*. They are built from simple, *atomic propositions* by using *propositional logical connectives*. The truth values propagate over all propositions through *truth tables* for the propositional connectives.

Propositional logic can only formalize simple logical reasoning that can be expressed in terms of propositions and their truth values, but it is quite insufficient for practical knowledge representation and reasoning. For that, it needs to be extended with several additional features, including *constants (names)* and *variables* for objects of any nature (numbers, sets, points, human beings, etc.), *functions and predicates* over objects, as well as *quantifiers* such as "for all objects $x(\ldots x \ldots)$," and "there exists an object x such that $(\ldots x \ldots)$." These lead to *first-order languages*, which (in many-sorted versions) are essentially sufficient to formalize most common logical reasoning. Designing appropriately expressive logical languages and using them to capture fragments of natural languages and reasoning is one of the main tasks of modern logic.

There are three major aspects of a logical system: *semantic*; *syntactic*; and *deductive*. The former deals mostly with the semantic notions of *truth, validity and logical consequence*, whereas the latter two deal respectively with the *syntax and grammar of logical languages* and with systems for *logical deduction* and *derivations and deductive consequences*. Deductive systems are purely mechanical procedures designed to *derive (deduce)* logical validities and consequences by means of formal *rules of inference* and possibly some postulated derived formulae called *axioms*. Thus, a deductive system does not refer explicitly to the meaning (semantics) of the formulae but only treats them as special strings of symbols and acts on their shape (syntax). In principle, a deductive system can be used successfully without any understanding of what formulae mean, and derivations in a deductive system can be performed not only by humans but also by artificial "agents" or computers. However, deductive systems are always meant to capture (or even determine) logical consequence so, ideally, semantic logical consequence and deductive consequence should precisely match each other. If that is the case, we say that the deductive system is *sound and complete*, or just *adequate*. Design and study of adequate and practically useful deductive systems is another major logical task.

The main syntactic, semantic, and deductive aspects of classical logic are discussed in detail in the book; there is much more that is not treated here however, both inside and outside of classical logic. In particular, logic is deeply related to: the foundations of mathematics, via *axiomatic theories of sets*; mathematics itself via *model theory*; the important notions of *algorithmic decidability and computability* via *recursion theory*; and the fundamentals and limitations of the deductive approach via *proof theory*. All of these are major branches of logic that I will only mention briefly in the text, but much more can be seen in the references. Furthermore, there is a rich variety of other, more specialized *non-classical* logical languages and systems that are better suited for specific modes and aspects of reasoning, such as *intuitionistic, modal, temporal, epistemic, deontic, and non-monotonic logics* that will not (except briefly intuitionistic logic) be discussed at all in this book. References to relevant publications covering these topics are provided throughout.

Finally, a few final words on the role of logic in the modern world. As I mentioned earlier, contemporary logic has become a highly interdisciplinary area with fundamental applications to a wide variety of scientific fields including mathematics, philosophy, computer science, artificial intelligence, and linguistics. Today logic not only provides methodology for correct human reasoning, but also techniques and tools for automated reasoning of intelligent agents. It also provides theoretical foundations for basic concepts in computer science such as *computation and computability*, *algorithms and complexity*, and *semantics of programming languages*, as well as practical tools for formal specification, synthesis, analysis, and verification of software and hardware, development and management of intelligent databases, and logic programming. The impact of logic on computer science nowadays is often compared to the impact of differential and integral calculus on natural sciences and engineering from the 17th century.

I end this introduction with a humble hope that this book will help the reader understand and master the use of this great intellectual tool called Logic. Enjoy it!

Valentin Goranko
Stockholm, November 2015

An Appetizer: Logical Paradoxes and Self-Reference

Dear reader,

The sentence that you are reading now is not true.

Is this claim true or false? If true, then it truly claims that it is not true, so it can't be true. But then, it is *not true* that it is not true, it *must* be true! Or ...?

This is a version of probably the oldest known *logical paradox* since antiquity, also known as the **liar's paradox** which refers to the quote "I am lying now."

What is a logical paradox? It is a statement or an argument that presents an apparent logical contradiction, either with well-known and accepted truths or simply with itself. Unlike a *fallacy*, a paradox is not due to an incorrect reasoning, but it could be based on wordplay or on a subtle ambiguity in the assumptions or concepts involved. Most commonly however, logical paradoxes arise when using *self-reference*, such as in the opening sentence above. Logicians love playing with self-reference. For instance, I have added this sentence in order to make a reference to itself. And, this one, which *does not* make a reference to itself. (Or, does it ...?)

A variation of the liar's paradox is **Jourdain's card paradox**, which does not rely on immediate self-reference but on a circular reference. Here is a simple version:

The next sentence is true. The previous sentence is false.

I end this appetizer two more paradoxes which are not exactly logical but *semantic*, again a self-referential play but now with natural language.

The first is known as **Berry's paradox.** Clearly every natural number can be defined in English with sufficiently many words. However, if we bound the number of words to be used, then only finitely many natural numbers can be defined. Then, there will be numbers that cannot be defined with that many words. Hence, there must be a *least* so undefinable natural number. Now, consider the following sentence "*The least natural number that is not definable in English with less than twenty words*." There is a uniquely determined natural number that satisfies this description, so it *is a definition* in English, right? Well, count how many words it uses.

The second is the **Grelling–Nelson paradox**. Divide all adjectives into two groups: **autological**, if and only if it describes itself, such as “English,” “short,” and “fourteen-letter;” and **heterological**, if and only if it does not describes itself, such as “wet,” “white,” and “long.” Now, is the adjective “heterological” autological or heterological?

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`;