

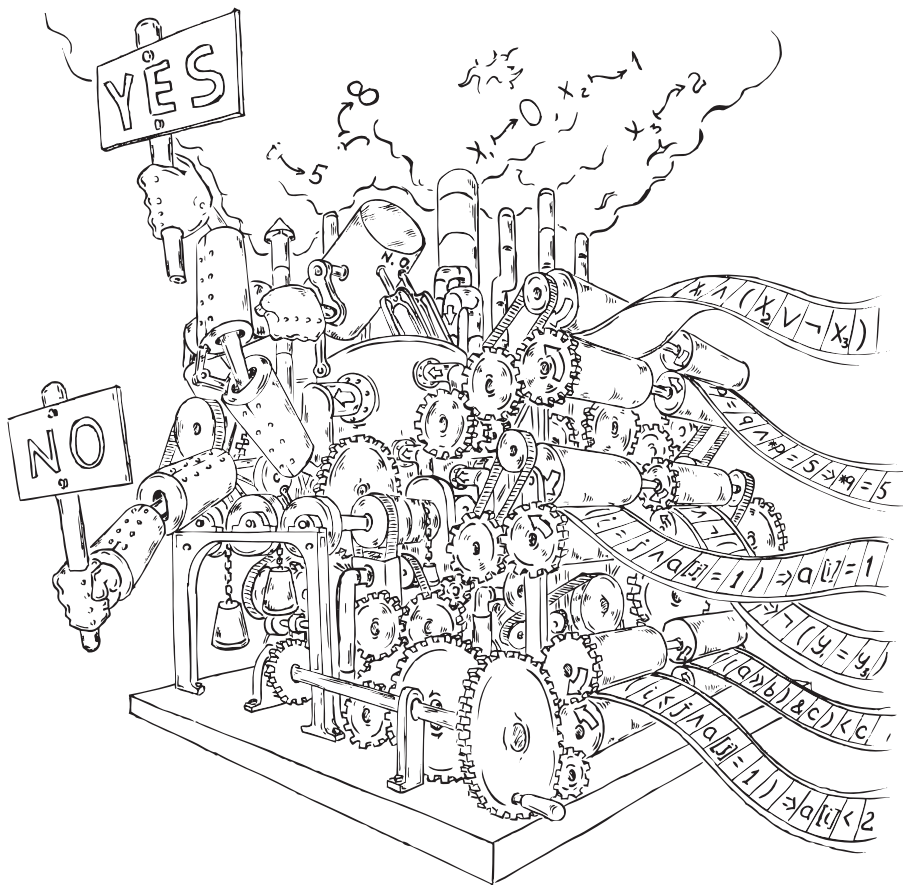
---

## Preface

A *decision procedure* is an algorithm that, given a decision problem, terminates with a correct yes/no answer. In this book, we concentrate on decision procedures for decidable first-order theories that are useful in the context of automated verification and reasoning, theorem proving, compiler optimization, synthesis, and so forth. Since the ability of these techniques to cope with problems arising in industry depends critically on decision procedures, this is a vibrant and prospering research subject for many researchers around the world, both in academia and in industry. Intel and AMD, for example, are developing and using theorem provers and decision procedures as part of their efforts to build circuit verification tools with ever-growing capacity. Microsoft is developing and routinely using decision procedures in several code analysis tools.

Despite the importance of decision procedures, one rarely finds a university course dedicated entirely to this topic; occasionally, it is addressed in courses on algorithms or on logic for computer science. One of the reasons for this situation, we believe, is the lack of a textbook summarizing the main results in the field in an accessible, uniform way. The primary goal of this book is therefore to serve as a textbook for an advanced undergraduate- or graduate-level computer science course. It does not assume specific prior knowledge beyond what is expected from a third-year undergraduate computer science student. The book may also help graduate students entering the field, as currently they are required to gather information from what seems to be an endless list of articles.

The decision procedures that we describe in this book draw from diverse fields such as graph theory, logic, and operations research. These procedures have to be highly efficient, since the problems they solve are inherently hard. They never seem to be efficient enough, however: what we want to be able to prove is always harder than what we *can* prove. Their asymptotic complexity and their performance in practice must always be pushed further. These characteristics are what makes this topic so compelling for research and teaching.



**Fig. 1.** Decision procedures can be rather complex ... those that we consider in this book take formulas of different theories as input, possibly mix them (using the Nelson–Oppen procedure – see Chap. 10), decide their satisfiability (“YES” or “NO”), and, if yes, provide a satisfying assignment

### Which Theories? Which Algorithms?

A first-order theory can be considered “interesting”, at least from a practical perspective, if it fulfills at least these two conditions:

1. The theory is expressive enough to model a real decision problem. Moreover, it is more expressive or more natural for the purpose of expressing some models in comparison with theories that are easier to decide.

2. The theory is either decidable or semidecidable, and more efficiently solvable than theories that are more expressive, at least in practice if not in theory.<sup>1</sup>

All the theories described in this book fulfill these two conditions. Furthermore, they are all used in practice. We illustrate applications of each theory with examples representative of real problems, whether they may be verification of C programs, verification of hardware circuits, or optimizing compilers. Background in any of these problem domains is not assumed, however.

Other than in one chapter, all the theories considered are quantifier-free. The problem of deciding them is NP-complete. In this respect, they can all be seen as “front ends” of any one of them, for example propositional logic. They differ from each other mainly in how naturally they can be used for modeling various decision problems. For example, consider the theory of equality, which we describe in Chaps. 3 and 4: this theory can express any Boolean combination of Boolean variables and expressions of the form  $x_1 = x_2$ , where  $x_1$  and  $x_2$  are variables ranging over, for example, the natural numbers. The problem of satisfying an expression in this theory can be reduced to a satisfiability problem of a propositional logic formula (and vice versa). Hence, there is no difference between propositional logic and the theory of equality in terms of their ability to model decision problems. However, many problems are more naturally modeled with the equality operator and non-Boolean variables.

For each theory that is discussed, there are many alternative decision procedures in the literature. Effort was made to select those procedures that are known to be relatively efficient in practice, and at the same time are based on what we believe to be an interesting idea. In this respect, we cannot claim to have escaped the natural bias that one has towards one’s own line of research.

Every year, new decision procedures and tools are being published, and it is impossible to write a book that reports on this moving target of “the most efficient” decision procedures (the worst-case complexity of most of the competing procedures is the same). Moreover, many of them have never been thoroughly tested against one another. We refer readers who are interested in the latest developments in this field to the SMT-LIB Web page, as well as to the results of the annual tool competition SMT-COMP (see Appendix A). The SMT-COMP competitions are probably the best way to stay up to date as to the relative efficiency of the various procedures and the tools that implement them. One should not forget, however, that it takes much more than a good algorithm to be efficient in practice.

## The Structure and Nature of This Book

The first chapter is dedicated to basic concepts that should be familiar to third- or fourth-year computer science students, such as formal proofs, the

---

<sup>1</sup> Terms such as *expressive* and *decidable* have precise meanings, and are defined in the first chapter.

satisfiability problem, soundness and completeness, and the trade-off between expressiveness and decidability. It also includes the theoretical basis for the rest of the book. From Sect. 1.5 onwards, the chapter is dedicated to more advanced issues that are necessary as a general introduction to the book, and are therefore recommended even for advanced readers. Each of the 10 chapters that follow is mostly self-contained, and generally does not rely on references to other chapters, other than the first introductory chapter. An exception to this rule is Chap. 4, which relies on definitions and explanations given in Chap. 3.

The mathematical symbols and notations are mostly local to each chapter. Each time a new symbol is introduced, it appears in a rounded box in the margin of the page for easy reference. All chapters conclude with problems, varying in level of difficulty, and bibliographic notes and a glossary of symbols.

A draft of this book was used as lecture notes for a combined undergraduate and graduate course on decision procedures at the Technion, Israel, at ETH Zurich, Switzerland, and at Oxford University, UK. The slides that were used in these courses, as well as links to other resources appear on the book's Web page ([www.decision-procedures.org](http://www.decision-procedures.org)). Source code of a C++ library for rapid development of decision procedures can also be downloaded from this page. This library provides the necessary infrastructure for programming many of the algorithms described in this book, as explained in Appendix B. Implementing one of these algorithms was a requirement in the course, and it proved successful. It even led several students to their thesis topic.

## Acknowledgments

Many people read drafts of this manuscript and gave us useful advice. We would like to thank, in alphabetical order, Domagoj Babic, Josh Berdine, Hana Chockler, Leonardo de Moura, Benny Godlin, Alan Hu, Wolfgang Kunz, Shuvendu Lahiri, Albert Oliveras Llunell, Joel Ouaknine, Hendrik Post, Sharon Shoham, Aaron Stump, Cesare Tinelli, Ashish Tiwari, Rachel Tzoref, Helmut Veith, Georg Weissenbacher, and Calogero Zarba. We thank Ilya Yodovsky Jr. for the drawing in Fig. 1.

February 2008

Daniel Kroening  
Oxford University, United Kingdom

Ofer Strichman  
Technion, Haifa, Israel



<http://www.springer.com/978-3-540-74104-6>

Decision Procedures

An Algorithmic Point of View

Kroening, D.; Strichman, O.

2008, XVI, 306 p., Hardcover

ISBN: 978-3-540-74104-6