https://doi.org/10.1007/s00165-018-0466-2

Formal Aspects of Computing (2018) 30: 759

Formal Aspects of Computing

CrossMark

Daniel Kroening and Ofer Strichman: Decision procedures

Springer Verlag, 2016, XXI,+356 ISBN 978-3-662-50496-3 (Hardback, €69,67), http://www.decision-procedures.org/

Rosemary Monahan

Department of Computer Science,
Maynooth University, Maynooth, Ireland

This is an excellent book, which I am delighted to have the chance to review. I have used the first edition of this book to introduce decision procedures to graduate and undergraduate students studying software verification techniques. The text and the supporting material have been invaluable, stepping the reader through decision procedures and their combinations.

The second edition offers both an updated introduction to the topic of decision procedures and more advanced material concerning topics such as quantification, efficiency in SAT solving and applications of SMT solving in industry. The authors focus on decision procedures for decidable first-order theories that are used primarily in automated software and hardware verification as well as theorem proving. Decision procedures are presented with clear definitions, algorithms and updated examples representative of real-world problems. It is suitable for advanced undergraduate students, MSc level graduates and research students.

I have read the book in detail and compared it to the first edition. Chapters 1, 2 and 3 provide an overview of basic concepts and material required to understand satisfiability. In this second edition, the focus has moved from the DPLL framework to Conflict-Driven Clause Learning (CDCP) based procedures for deciding propositional formulae and more detail is provided on SAT solvers and the constraint satisfaction problem (SCP). This is a sensible update as it coincides with advances in tools for SAT and SMT solving. Chapter 3 provides an extended coverage of DPLL(T), a generalisation of CDCL to a decision procedure for decidable quantifier-free first-order theories. This is updated and made central to the text from the first edition, where DPLL(T) was discussed in a subsection of Chapter 11. This update is welcome as DPLL(T) is implemented in many current SMT solvers. As before chapter 4–10 are self contained, covering equality logic, uninterpreted functions, linear arithmetic, bit vectors, arrays, pointer logic, quantified formulae and the decidability of their combinations via the Nelson–Oppen Combination Procedure. Chapter 11 discusses lazy and eager encodings, focusing on eager approaches for eliminating uninterpreted functions via a reduction to equality logic constraints. Graph based methods for an eager encoding of equality logic formulae into propositional logic are also described. This chapter reorganises material from the first edition of the book, bringing it together to discuss the topic as an alternative to the DPLL(T) approach. Chapter 12 offers new material which describes industrial applications in Software Engineering and Computational Biology, with sections contributed from researchers at Microsoft Research. Topics presented, and supporting examples, are interesting for both researchers and students learning about SAT and SMT, covering topics such as bounded and unbounded program analysis, DNA computing and gene regulatory networks. A short SMT-Lib tutorial (in the appendices) is a further welcome addition to the text.

In summary, I highly recommend the book to those interested in understanding and contributing to the world of SAT and SMT solving and hope that the teaching materials offered alongside the first edition will be updated to accompany the new edition.

Published online 2 August 2018

Correspondence and offprint requests to: R. Monahan, E-mail: Rosemary.Monahan@mu.ie