Gregor Engels, Regina Hebig, Matthias Tichy (Hrsg.): SE 2023,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2023 129

Variational Satisfiability Solving: Efficiently Solving Lots of Related SAT Problems – Summary

Jeffrey M. Young[1] Paul Maximilian Bittner[2] Eric Walkingshaw[3] Thomas Thüm[4]

Abstract: We report about recent research on satisfiability solving for variational domains, originally published in 2022 in the Empirical Software Engineering Journal (EMSE) within the special issue on configurable systems [Yo22]. Incremental SAT solving is an extension of classic SAT solving that enables solving a set of related SAT problems by identifying and exploiting shared terms. However, using incremental solvers effectively is hard since performance is sensitive to the input order of subterms and results must be tracked manually. This paper translates the ordering problem to an encoding problem and automates the use of incremental solving. We introduce *variational* SAT solving, which differs from incremental solving by accepting all related problems as a single variational input and returning all results as a single variational output. Variational SAT solving automates the interaction with the incremental solver and enables a method to automatically optimize sharing in the input. We formalize a variational SAT algorithm, construct a prototype variational solver, and perform an empirical analysis on two real-world datasets that applied incremental solvers to software evolution scenarios. We show that the prototype solver scales better for these problems than four off-the-shelf incremental solvers while also automatically tracking individual results.

Keywords: satisfiability solving, variation, choice calculus, software product lines

Summary

Satisfiability (SAT) solving is a ubiquitous technology in software product lines for a diverse set of analyses ranging from anomaly detection, dead code analysis, sampling, and automated analysis of feature models. The general pattern is to represent parts of the system or feature model as a propositional formula, and reduce the analysis to a SAT problem. However, modern software is constantly evolving and thus the translation step to a single SAT problem quickly becomes a translation to a set of SAT problems.

Incremental SAT solvers allows the user to hand-write a program to optimize for sharing of equal subterms in a set of related SAT problems to solve. Theoretically, this is more efficient because it reuses knowledge of shared terms. However, using an incremental solver in this way requires substantial manual effort and domain knowledge, produces a specific solution to a specific analysis, and it requires extra infrastructure to manage results.

[1] IOHK, Longmont, Colorado, jeffrey.young@iohk.io
[2] University of Ulm, Germany, paul.bittner@uni-ulm.de
[3] Unaffiliated, Corvallis, OR, USA, ewalkingshaw@acm.org
[4] University of Ulm, Germany, thomas.thuem@uni-ulm.de

In this paper, we show that the performance benefit of using an incremental solver to solve a large number of related SAT problems can be achieved while mitigating the main drawbacks of incremental solvers, that is, high manual effort and deep integration with the application. Our solution is to formalize a method of *variational* SAT solving that makes use of known commonalities among propositional formulas and automates the interaction with an incremental solver. Variational SAT solving takes as input a single *variational formula*, which encodes a set of related SAT problems to solve. It then compiles this variational formula into an efficient program to run on an incremental solver. Finally, it collects the results from the incremental solver into a single result that captures the solutions to all of the SAT problems described by the input variational formula.

Our approach has several benefits: (1) End-users are only required to provide a single variational formula, which represents a set of related propositional formulas, rather than a formula *and* a hand-written program to direct the solver. (2) It is general; while variational satisfiability solving is applied to feature model analyses in this work, it can be used for any analysis that can be encoded as a variational formula, which can be constructed from any set of related SAT problems. (3) With a variational formula, new kinds of syntactic manipulations and optimizations, such as factoring out shared terms, become possible and can be automated. (4) A *variational model* may be produced that encapsulates a set of satisfying assignments for all variants of the variational formula, alleviating the need to track the incremental solver's results when satisfying assigments are needed.

Data Availability

The original publication is accepted for submission and our camera-ready copy is currently processed by the publisher. A preprint is available online at `https://github.com/SoftVarE-Group/Papers/raw/main/2022/2022-EMSE-Young.pdf`. Our artifact is available on Github (`https://github.com/doyougnu/VSat/tree/EMSE-revision-1`) and Zenodo (DOI: 10.5281/zenodo.5543884). Our evaluation data is available on Zenodo (DOI: 0.5281/zenodo.5546009).

Bibliography

[Yo22] Young, Jeffrey M.; Bittner, Paul Maximilian; Walkingshaw, Eric; Thüm, Thomas: Variational Satisfiability Solving: Efficiently Solving Lots of Related SAT Problems. Empirical Software Engineering (EMSE), 28, November 2022.