(Hrsg.): Software Engineering 2024,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2024 11

Evaluating State-of-the-Art #SAT Solvers on Industrial Configuration Spaces

Chico Sundermann[1] Tobias Heß[1] Michael Nieke[2] Paul Maximilian Bittner[1] Jeffrey M. Young[3] Thomas Thüm[1] Ina Schaefer[4]

Abstract: We report about recent research on model counting in the context of configurable software, published at the Empirical Software Engineering Journal (EMSE) [Su23].

Product lines are widely used to manage families of products that share a common base of features. Typically, not every combination (configuration) of features is valid. Feature models are a de facto standard to specify valid configurations and allow standardized analyses on the variability of the underlying system. A large variety of such analyses depends on computing the number of valid configurations. To analyze feature models, they are typically translated to propositional logic. This allows to employ #SAT solvers that compute the number of satisfying assignments of the propositional formula translated from a feature model. However, the #SAT problem is generally assumed to be even harder than SAT and its scalability when applied to feature models has been explored only sparsely. We empirically evaluate 21 publicly available #SAT solvers on 130 feature models. Our results indicate that current solvers master a majority of the considered feature models with the fastest solvers requiring less than one second for each successfully evaluated feature model. However, there are two complex systems for which none of the evaluated solvers scales.

Keywords: feature modeling, product lines, model counting, #SAT

Summary

Product lines are commonly used to develop, test, and evolve a family of similar products to reduce the costs compared to developing each product separately. Each product in a product line is composed from a set of reusable *features*, which are generally shared across multiple products. Typically, not every combination of selected features (i. e., a *configuration*) is valid and results in a functional product. To specify the set of valid configurations for a given product line, engineers can use *feature models*. A feature model consists of a set of features and a set of *constraints* limiting the valid configurations.

Manually keeping track of all constraints is infeasible, as industrial feature models may contain thousands of features and hundreds of thousands of constraints. Automated reasoning is typically used to analyze feature models, for example to check whether a given configuration

[1] University of Ulm, Germany, firstname.surname@uni-ulm.de
[2] Technische Universität Braunschweig – Brunswick, Germany, michael.nieke@mailbox.org
[3] IOHK – Longmont, Colorado, jeffrey.young@iohk.io
[4] Karlsruhe Institute of Technology, Germany, ina.schaefer@kit.edu

is valid (i. e., it conforms to all constraints imposed by the feature model). Consequently, a large variety of automated support in terms of analyses has been proposed. A multitude of analyses is based on feature-model counting (i. e., computing the number of valid configurations).

The scalability of available tools, that enable counting, on industrial feature models is largely unknown. Here, we focus on propositional model counting (i.e., #SAT). As the translation of feature models to propositional logic is well-researched, #SAT solvers can be applied out of the box to compute the cardinality of feature models. #SAT, however, is a computationally complex problem. While it is widely accepted that regular SAT is typically easy for industrial feature models (compared to randomly generated formulas), this has not been explored for #SAT.

In this work, we provide insights on the scalability of modern off-the-shelf #SAT solvers for the analysis of feature models. Analyses based on feature-model counting can only be applied in practice if available #SAT solvers scale to industrial feature models considering time restrictions for typical use cases, such as interactive settings or continuous integration environments. We thus evaluate the runtimes of analyzing feature models with publicly available #SAT solvers. Hereby, we also examine different strategies for #SAT solvers. Overall, our work provides the following contributions:

1. We examine the runtime of #SAT technology on 130 industrial feature models.
2. We identify best performing #SAT solvers out of 21 off-the-shelf tools.
3. We compare the benefits of different #SAT technologies.
4. We examine the correlation between the runtime of #SAT solvers and structural metrics of the feature model.
5. We inspect the performance of two approximate #SAT solvers.
6. We provide the number of valid configurations for feature models in our dataset.

Data Availability

The article is available at `https://link.springer.com/article/10.1007/s10664-022-10265-9`. A preprint is also available online at `https://github.com/SoftVarE-Group/Papers/blob/main/2023/2023-EMSE-Sundermann.pdf`. The datasets generated during and/or analyzed during the current study are available in the replication repository, `https://zenodo.org/records/7329979`.

Bibliography

[Su23] Sundermann, Chico; Heß, Tobias; Nieke, Michael; Bittner, Paul Maximilian; Young, Jeffrey M.; Thüm, Thomas; Schaefer, Ina: Evaluating State-of-the-Art #SAT Solvers on Industrial Configuration Spaces. Empirical Software Engineering (EMSE), 28, 2023.