Family-Based Model Checking using Off-the-Shelf Model Checkers

[Extended Abstract]

Aleksandar S. Dimovski
IT University of Copenhagen
adim@itu.dk

Ahmad Salim Al-Sibahi
IT University of Copenhagen
asal@itu.dk

Claus Brabrand
IT University of Copenhagen
brabrand@itu.dk

Andrzej Wąsowski
IT University of Copenhagen
wasowski@itu.dk

1. ABSTRACT

Model checking provides a convenient way to check whether a given software system is correct with respect to a set of relevant semantic properties. To use a *model checker* like SPIN [5], the software system must be modelled as a transition system (TS). Afterwards, the model checker can check the correctness of the translated TS by exhaustively exploring all possible transitions.

For families of software systems Classen et al. [1] present a *lifted* model checker SNIP, where each family is modelled as a Featured TS [2] that has transitions guarded by feature expressions. SNIP is highly specialized and uses heuristics to avoid naïvely iterating through all possible variations; however, the number of configurations is still exponential in size and thus the model checker can only feasibly handle systems of a limited size.

We adapt our previous work on applying variability abstraction to lifted data-flow analysis [3] to the setting of lifted model checking. We present a calculus of variability abstractions that trade precision for speed while preserving correctness [4]. The abstractions work symbiotically with the lifted model checker SNIP, but can also work with the classical and efficient off-the-shelf model checker SPIN without requiring any knowledge of variability. We prove semantically how each abstraction operation in the calculus forms a Galois collection, and therefore is suitable to use in abstract interpretation of Featured TS. Furthermore, we present an equivalent lightweight syntactic transformation tool that works directly on the input text files and does not require explicitly constructing the corresponding Featured TS in memory.

Our results show that there are orders of magnitudes to be gained in performance compared to performing lifted analysis alone; we show how our tool scales better than the existing tools and makes analysing some previously infeasible models feasible. Furthermore, we also show that many models could be verified swiftly using the abstracted analysis without requiring all of the precision that a concrete analysis provides.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SPLC '15 July 20-24, 2015, Nashville, TN, USA

© 2015 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3613-0/15/07.

DOI: 10.1145/2791060.2791119

CCS Concepts

•**Theory of computation** → **Program verification; Abstraction;**

Keywords

Software Product Lines; Model Checking; Abstract Interpretation

Acknowledgements

This work has been funded by the Danish Council for Independent Research under the Sapere Aude scheme, grant no. 0602-02327B.

2. REFERENCES

[1] A. Classen, M. Cordy, P. Heymans, A. Legay, and P. Schobbens. Model checking software product lines with SNIP. *STTT*, 14(5):589–612, 2012.

[2] A. Classen, M. Cordy, P. Schobbens, P. Heymans, A. Legay, and J. Raskin. Featured transition systems: Foundations for verifying variability-intensive systems and their application to LTL model checking. *IEEE Trans. Software Eng.*, 39(8):1069–1089, 2013.

[3] A. Dimovski, C. Brabrand, and A. Wąsowski. Variability abstractions: Trading precision for speed in family-based analyses. In *ECOOP 2015 - Object-Oriented Programming - 29th European Conference*, 2015. To Appear.

[4] A. S. Dimovski, A. S. Al-Sibahi, C. Brabrand, and A. Wąsowski. Family-based model checking without a family-based model checker. In *Model Checking Software, 22nd International SPIN Workshop, Stellenbosch, South Africa, August 24 - 26, 2015*, 2015. To Appear.

[5] G. J. Holzmann. *The SPIN Model Checker - primer and reference manual*. Addison-Wesley, 2004.