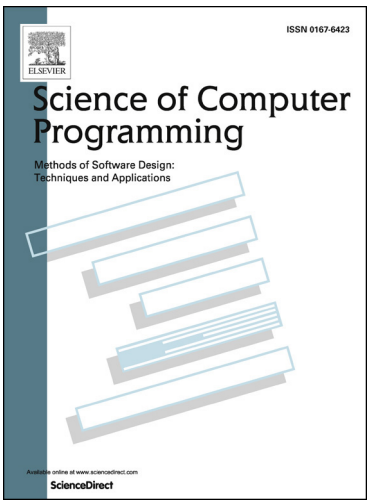
Accepted Manuscript



Variability abstractions for lifted analyses

Aleksandar S. Dimovski, Claus Brabrand, Andrzej Wasowski

PII: S0167-6423(18)30058-3
DOI: https://doi.org/10.1016/j.scico.2017.12.012
Reference: SCICO 2192

To appear in: *Science of Computer Programming*

Received date: 16 December 2016
Revised date: 28 October 2017
Accepted date: 8 December 2017

Please cite this article in press as: A.S. Dimovski et al., Variability abstractions for lifted analyses, *Sci. Comput. Program.* (2018), https://doi.org/10.1016/j.scico.2017.12.012

Variability abstractions for lifted analyses[☆]

Aleksandar S. Dimovski[a,b,*], Claus Brabrand[b], Andrzej Wasowski[b]

[a] *Mother Teresa University, 12 Udarna Brigada 2a, 1000 Skopje, Makedonija*
[b] *IT University of Copenhagen, Rued Langgaards Vej 7, 2300 Copenhagen, Denmark*

Abstract

Family-based (lifted) static analysis for "highly configurable programs" (program families) is capable of analyzing all variants at once without generating any of them explicitly. It takes as input only the common code base, which encodes all variants of a program family, and produces precise analysis results corresponding to all variants. However, the computational cost of the lifted analysis still depends inherently on the number of variants, which is in the worst case exponential in the number of statically configurable options (features). For a large number of features, the lifted analysis may be too costly or even infeasible. In this work, we introduce variability abstractions defined as Galois connections, which simplify variability away from program families based on `#ifdef`-s. Then, we use abstract interpretation as a formal method for the calculational-based derivation of abstracted lifted analyses, which are sound by construction.

Our approach for abstracting lifted analysis is orthogonal to the particular program analysis chosen as a client. While a single program analysis operates on program states and depends on language-specific constructs, the lifted analysis assumes that a single program analysis already exists and lifts its results to all variants of the analyzed program family. Variability abstractions aim to reduce this variability-specific component of the lifted analysis, which handles variability and `#ifdef`-s. Furthermore, given the "orthogonality" of variability abstractions to the rest of the analysis (its language-specific component), we can implement abstractions as a preprocessor. In particular, given an abstraction we define a syntactic transformation, which translates any program family into an abstracted version of it, such that the analysis of the abstracted program family coincides with the corresponding abstracted analysis of the original program family. We have implemented the proposed approach, and we evaluate its practicality on three Java benchmarks. The evaluation shows that abstractions yield significant performance gains, especially for families with higher variability.

Keywords: Program Families, Static Analysis, Abstract Interpretation

[☆]Partially supported by The Danish Council for Independent Research under a Sapere Aude project, VARIETE.

[*]Corresponding author