Towards Computing Abstract Distances in Logic Programs*

Ignacio Casso
IMDEA Software Institute
Madrid, Spain
Universidad Politcnica de Madrid (UPM)
Madrid, Spain
ignacio.decasso@imdea.org

Jose F. Morales
IMDEA Software Institute
Madrid, Spain
josef.morales@imdea.org

Pedro Lopez-Garcia
IMDEA Software Institute
Madrid, Spain
Spanish Council for Scientific Research (CSIC)
Madrid, Spain
pedro.lopez@imdea.org

Manuel V. Hermenegildo
IMDEA Software Institute
Madrid, Spain
Universidad Politcnica de Madrid (UPM)
Madrid, Spain
manuel.hermenegildo@imdea.org

Many practical static analyzers are based on the theory of Abstract Interpretation. The basic idea behind this technique is to interpret (i.e., execute) the program over a special abstract domain D_α to obtain some abstract semantics S_α of the program P, which will over-approximate every possible execution of P in the standard (concrete) domain D. This makes it possible to reason safely (but perhaps imprecisely) about the properties that hold for all such executions.

When designing or choosing an abstract interpretation-based analysis, a crucial issue is the trade-off between cost and accuracy, and thus research in new abstract domains, widenings, fixpoints, etc., often requires studying this trade-off. However, while measuring analysis cost is typically relatively straightforward, having effective accuracy metrics is much more involved. There have been a few proposals for this purpose, including, e.g., probabilistic abstract interpretation and some metrics in numeric domains, but they have limitations and in practice most studies come up with ad-hoc accuracy metrics, such as counting the number of program points where one analysis is strictly more precise than another.

We propose a new approach for measuring the accuracy of abstract interpretation-based analyses in (C)LP. It is based on defining distances in abstract domains, denoted *abstract distances*, and extending them to distances between inferred semantics or whole analyses of a given program, over those domains. The difference in accuracy between two analyses can then be measured as the distance between them, and the accuracy of an analysis can be measured as the distance to the actual abstract semantics, if known.

We first develop some general theory on metrics in abstract domains. Two key points to consider here are the structure of an abstract domain as a lattice and the relation between the concrete and abstract domains. With regard to the first point, we survey and extend existing theory and proposals for distances in a lattice L. The distances are often based in a partial distance $d_\sqsubseteq : \{(a,b) \mid (a,b) \in L \times L,\ a \sqsubseteq b\} \to R$ between related elements of the lattice, or in a monotonic size $size : L \to R$. With regard to the second, we study the relation between distances d and d_α in the concrete and abstract domains, and the abstraction and concretization functions $\alpha : D \to D_\alpha$, $\gamma : D_\alpha \to \alpha$. In that sense we observe that both α and γ induce distances $d^\gamma_\alpha : D_\alpha \times D_\alpha \to R$, $d^\gamma_\alpha(a,b) = d(\gamma(a),\gamma(b))$ and $d^\alpha : D \times D \to R$, $d^\alpha(A,B) = d_\alpha(\alpha(A),\alpha(B))$ in the abstract and concrete domains from distances d and d_α in the concrete and abstract domains respectively.

*This document is an extended abstract of Technical Report CLIP-2/2019.0 [1]. Research partially funded by MINECO project TIN2015-67522-C3-1-R *TRACES* and Comunidad de Madrid project S2018/TCS-4339 *BLOQUES-CM*, co-funded by EIE Funds of the European Union.

Submitted to:
ICLP 2019

We then build on this theory and ideas in order to propose metrics for a number of common domains used in (C)LP analysis. In particular we define formally distances for the domains `share` and `regular types`, and show that they are indeed metrics. The domain *share* abstracts information about variable sharing between terms in a substitutions, and the distance there builds on a notion of *size* in the domain, based on the set-based structure of the domain. The *regular types* domain abstracts information about the shape of the terms in a substitution, and the distance there is based on the abstraction of a Hausdorff distance between sets of terms in the concrete domain.

We then extend these metrics to distances between abstract interpretation-based analyses of a whole program, that is, distances in the space of AND-OR trees that represent the abstract execution of a program over an abstract domain. We proposed three extensions in increasing order of complexity. The first is the *top* distance, which only considers the roots of the AND-OR trees, i.e., the top result or abstract answer of the analysis, and computes the abstract distance between the abstract substitutions in those roots. The second is the *flat* distance, which groups together all nodes of the tree corresponding to the same program point, by means of the *least upper bound operation*, and is based on the abstract distances between the resulting abstract substitutions in each program point. The third is the *tree distance*, which considers the whole tree as a whole, computing the abstract distances node to node, and thus it is a metric. All these distances between analyses are thus parametric on an abstract distance in the underlying abstract domain.

These distances can then be used to compare quantitatively the accuracy of different abstract interpretation-based analyses of a whole program, by just calculating the distances between the representation of those analyses as AND-OR trees. This results in a principled methodology to measure differences of accuracy between analyses, which can be used to measure the accuracy of new fixpoints, widenings, etc. within a given abstract interpretation framework, not requiring knowledge of its implementation (i.e., apart from the underlying domain, everything else can be treated as a black box, if the framework provides a unified representation of analysis results as AND-OR trees).

Finally, we implement the proposed distances within the CiaoPP framework [2] and apply them to study the accuracy-cost trade-off of different sharing-related (C)LP analyses over a number of benchmarks and a real program. The domains *share-free*, *share*, *def*, and *share-free clique*, with a number of widenings, are used for this experiment. For the accuracy comparison, all the analyses results are translated so as to be expressed in terms of a common domain, *share* (i.e., their accuracy is compared only with respect of the sharing information they infer), and the loss of accuracy for each one is computed as the distance to a most precise analysis computed as the "intersection" between all of them. The results align with our a-priori knowledge, confirming the appropriateness of the approach, but also allow us to obtain further useful information and insights on where to use each domain. These preliminary results lead us to believe that this application of distances is promising in a number of contexts such as debugging the accuracy of analyses or calibrating heuristics for combining different domains in portfolio approaches.

References

[1] I. Casso, J. F. Morales, P. Lopez-Garcia & M. V. Hermenegildo (2019): *Computing Abstract Distances in Logic Programs*. Technical Report CLIP-2/2019.0, The CLIP Lab, IMDEA Software Institute and T.U. Madrid. Available at `http://arxiv.org/abs/1907.13263`.

[2] M. Hermenegildo, G. Puebla, F. Bueno & P. Lopez Garcia (2005): *Integrated Program Debugging, Verification, and Optimization Using Abstract Interpretation (and The Ciao System Preprocessor)*. Science of Computer Programming 58(1–2), pp. 115–140.