

# Mixing Polyedra and Boxes Abstract Domain for Constraint Solving

Marie Pelleau<sup>1,2</sup>, Emmanuel Rauzy<sup>1</sup>, Ghiles Ziat<sup>2</sup>, Charlotte Truchet<sup>3</sup>, and Antoine Miné<sup>2</sup>

<sup>1</sup> École Normale Supérieure, France

<sup>2</sup> Université Pierre et Marie Curie, France

<sup>3</sup> IRISA, INRIA Rennes Bretagne Atlantique, France

In Constraint Programming (CP), the variables domains are always considered in isolation; any relation between the variables is expressed at the constraint level, and not in the domains. Hence, the search space is always a Cartesian product of some base set (interval, set of integer values) approximating the solution space.

Abstract Interpretation (AI) [1] is a general theory of approximation, often used for program analysis, which is not limited to Cartesian abstractions, but can also internalize relations between variables. Abstract domains with widely different expressiveness have been described (boxes, polyhedra, congruences, etc.), as well as methods to combine them.

We present a solving method combining Cartesian CP domains with polyhedra from AI [3]. Our method is implemented in AbSolute, a constraint solver based on abstract domains. Abstract consistency and split (or instantiation) operators are defined accordingly [2]; we retrieve classic hull or arc consistencies when applied to Cartesian abstract domains, but obtain new ones when applied to relational domains such as polyhedra. Upon this, AbSolute implements a generic solving method, defined as a classic iteration of propagation and split using the abstract operations.

The key ingredient in our work is a form of *reduced product* [1], a construction to derive more expressive abstract domains by combining existing ones and letting them exchange information. In particular, AbSolute supports both the polyedra domain [3], which can exactly represent linear constraints, and boxes that have a more general handling of non-linear constraints. A box-polyedra reduced product combines the best of both. Additionally, AbSolute uses a non-standard reduced product to improve the efficiency for CP problems: we classify constraints, dispatch them to the most relevant domain only, and stop propagating constraints once they have been fully internalized in some abstract domain (e.g., a linear constraint represented exactly in a polyhedron). Specific heuristics deal with the propagation loop and the variable choice.

We demonstrate experimentally that our method is able to successfully combine the efficiency of linear programming and the expressiveness of generic constraints. Figure 1 illustrates the reduction in domains to represent a solution by allowing more expressive domains combining boxes and polyhedra. We tested our prototype with good results on the Coconut benchmark

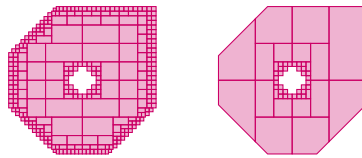


Fig. 1. Box solving on the left, mixed boxes and polyhedron on the right

## References

1. P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL'77*, pages 238–252. ACM, 1977.
2. M. Pelleau, A. Miné, C. Truchet, and F. Benhamou. A constraint solver based on abstract domains. In *VMCAI'13*, LNCS, page 17. Springer, Jan. 2013.
3. P. Cousot and N. Halbwachs. Automatic discovery of linear restraints among variables of a program. In *Conf. Rec. of the 5th Annual ACM SIGPLAN-SIGACT Symp. on Principles of Prog. Lang.*, pages 84–97, 1978. ACM Press.