

PASO: Proof, Static Analysis and Optimisation

ABSTRACT

The project PASO, funded by DIGITEO, is devoted to the analysis and proof of numerical properties of programs, arising in particular from the modeling of complex systems with critical security issues. It gathers computer scientists from CEA-LIST/MeASI, INRIA Saclay/Typical & LIX and specialists from Optimisation or Control theory from LIX/MeASI, INRIA Saclay/Maxplus & CMAP, and Supelec/L2S. The goal of this exploratory project is to cross-fertilise these fields, by applying advanced algorithms or techniques inspired by global optimization, by the analysis and identification of dynamical systems, or by zero-sum game theory, in order to improve the precision or the scalability of current methods in proof and static analysis. These applications coming from computer science turn out to raise new challenges for the applied mathematicians.

We present here a choice of recent works from some members of the project.

1. Formal Proofs and Optimization Problems (A. Mahboubi, B. Werner; INRIA Saclay/Typical & LIX)

Werner; INRIA Saclay/Typical & LIX

The use of computers has deeply changed the habits of the mathematical community. But will computers revolutionise the way mathematicians regard *proofs* themselves?

A *proof assistant* is a computer program implementing a logical formalism — the symbolic formulation of mathematical building blocks and the rules for combining them. The program guarantees that the steps of the *formal* proof are valid.

As an example, we can formalise the correctness proof of a computer program to guarantee that the program meets its specifications and thus consists of reliable code.

Optimization problems featuring intensive computations can rely on non-trivial correctness arguments and could benefit from a formal correctness check. A famous example is the effort initiated by T. Hales to provide a formal account of his proof of the Kepler conjecture (1998), which relies on a 40.000 lines C++ program implementing optimization techniques.

Kepler's conjecture
on optimal sphere packing



An inequality
from Hales' proof

$$\forall x_1, x_2, x_3, x_4, x_5, x_6 \in \mathbb{R},$$

$$\left[\left(\frac{2}{51} \right)^2 \leq x_1 \leq \left(\frac{2}{696} \right)^2 \wedge 4 \leq x_2 \leq \left(\frac{2}{168} \right)^2 \wedge \right.$$

$$4 \leq x_3 \leq \left(\frac{2}{168} \right)^2 \wedge 4 \leq x_4 \leq \left(\frac{2}{51} \right)^2 \wedge$$

$$4 \leq x_5 \leq \left(\frac{2}{51} \right)^2 \wedge 4 \leq x_6 \leq \left(\frac{2}{51} \right)^2 \implies$$

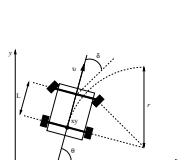
$$\frac{\pi}{50} < \pi + \arctan \frac{(x_1 x_3 + x_2 x_5 - x_1 x_4 - x_3 x_6 + x_4 x_6 - x_2(-x_2 + x_1 + x_3 - x_5 + x_4 + x_6))}{4x_2 x_2 x_5(-x_2 + x_1 + x_3 - x_5 + x_4 + x_6) + x_1 x_4(x_2 - x_1 + x_3 + x_5 - x_4 + x_6) + x_3 x_6(x_2 + x_1 - x_3 + x_5 + x_4 - x_6) - x_1 x_3 x_5 - x_2 x_3 x_4 - x_2 x_1 x_6 - x_5 x_4 x_6}$$

The Coq proof assistant, developed at INRIA Saclay, is well-equipped for this kind of task, because its objects are themselves programs, and computation is inherent to its logic. Furthermore, recent efforts have made computations inside the system more efficient by importing programming language technology.

Work is under way to pursue this advantage, in particular by providing safe but efficient numerical computations and developing optimization programs and libraries inside the system.

2 Planning of robust and reliable trajectories (M. Kieffer, E. Walter, Supelec/LSS)

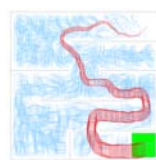
Supelec/LSS



$$\dot{x} = v \cos \theta$$

$$\dot{y} = v \sin \theta$$

$$\dot{\theta} = \frac{v}{L} \tan \delta$$



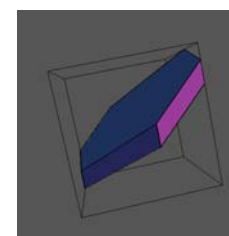
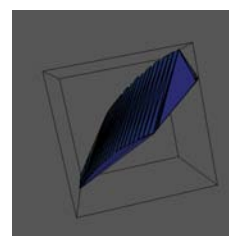
We look for a control such that for all perturbations with prescribed bounds, the trajectory, starting from a prescribed initial condition, avoids a forbidden region and reaches a target. This is achieved when the uncertainty (represented by boxes) remains limited, by using rapidly-exploring random trees and guaranteed numerical integrators.

3 New domains in static analysis (E. Goubault, S. Putot, CEA-LIST/MeASI)

We develop a version adapted to static analysis of the affine forms introduced by Comba et Stolfi in 1993. These forms describe the sets of values taken by affine combinations of finitely many independent "noise symbols" taking their values in the interval $[-1, 1]$. The concretisations of such forms in terms of subsets of \mathbb{R}^n yield a classical class of polyhedra with a central symmetry called *zonotopes*. However, the explicit parametrisation by affine forms shows a strong analogy with the Taylor methods which are used to perform guaranteed numerical computations or to analyse hybrid systems.

The best interpretation, both sound and accurate, of functions and programs by affine forms relies on partial order techniques combined with numerical approximation and optimisation methods like semidefinite programming relaxations.

The next two pictures show the inclusion of an algebraic variety in a zonotope which provides an approximation by affine forms of intervals, which is optimal in a well defined sense. The final picture shows a max-plus or tropical polytope, which offers a new promising domain to deal with disjunctive constraints, as shown in a recent work by X. Allamigeon (EADS), S. Gaubert and E. Goubault.



4. The dictionary between game theory and static analysis (S. Gaubert, E. Goubault, S. Putot, CEA-LIST/MeASI and INRIA Saclay/Maxplus & CMAP)

S. Gaubert, E. Goubault, S. Putot, CEA-LIST/MeASI and INRIA Saclay/Maxplus & CMAP

Zero-sum games with negative (!) discount
dynamical system
Shapley operator
horizon n problem
limit of the value in horizon n
value iteration

Abstract interpretation
program
functional
execution of n logical steps
optimal invariant (bound)
Kleene iteration

```
void main() {
    i = 1; j = 10;          i ≥ 1
    while (i <= j){ //1   j ≤ 10
        i = i + 2;         i ≤ j
        j = j - 1; }      i + 2j ≤ 21 (Invariants at
                        i + 2j ≥ 21 breakpoint 1)
```

To show this, we solve the following zero-sum game problem (by policy iteration):

$$\gamma(p) = ((1, 10) \cdot p) \vee (\bar{\gamma}(p) + (2, -1) \cdot p), \quad \forall p \in \mathcal{P} \setminus \{e_1 - e_2\}$$

$$\gamma(e_1 - e_2) = 0 \wedge (-9 \vee (\bar{\gamma}(e_1 - e_2) - 3)), \quad \bar{\gamma} = \text{convex hull}(\gamma), \quad \mathcal{P} = \{\pm e_1, \pm e_2, e_1 - e_2, \pm(e_1 + 2e_2)\}$$

5 Mathematical Programming (Leo Liberti, LIX/MeASI)

In static code analysis by abstract interpretation, the solution of the semantic equations describing the action of the computer program on some given domain is usually computed using Kleene's iteration method, suitably modified to converge to a least, or at least small, fixed point with respect to the domain lattice. We propose an alternative solution method based on modelling the semantic equations as a set of constraints in a mathematical program, with an objective function designed to select the smallest fixed point in the domain of intervals. For some classes of computer programs, the corresponding mathematical program is a Mixed-Integer Linear Program, which we solve to *guaranteed optimality* using a general-purpose Branch-and-Bound solver. With non-affine arithmetic the model belongs to the class of Mixed-Integer Nonlinear Programs, which are much more difficult to solve to optimality, although suboptimal feasible solutions can be found relatively easily by means of the Variable Neighbourhood Search algorithm. Our mathematical programming formulation guarantees that suboptimal feasible solutions represent valid fixed points (although not least) of the semantic equations, which implies that we can find *at least some* fixed points for codes involving difficult-to-treat non-affine arithmetics.