Thoughts on Trusting RAHD Computations

Grant Olney Passmore and Paul B. Jackson, Edinburgh

1 Motivation

Methods for deciding quantifier-free nonlinear arithmetical conjectures over $\mathbb{R}$ are crucial in the formal verification of many real-world systems. Though quantifier-free nonlinear real arithmetic is decidable, it is infeasible: any general decision method for this problem must be worst-case exponential in the number of variables (dimension) of the input formula. This is unfortunate, as many important properties of systems are naturally modelled by high-dimensional conditions. Despite this infeasibility, many different decision methods have been developed, each with their own strengths and weaknesses. Moreover, arithmetical verification conditions arising from real-world systems often have nice properties (such as low-degree non-linearity) making them amenable to restricted variants of decision methods which are more efficient than their general counterparts. RAHD is a proof procedure for real arithmetic which works to combine a heterogeneous collection of real algebraic decision methods so as to exploit their respective "sweet spots."

2 Trust

RAHD provides original implementations of many decision techniques for fragments of the elementary theory of real closed fields. These include quantifier elimination (qe) by Muchnik sequences, qe by quadratic virtual term substitution, qe by full-dimensional extended partial cylindrical algebraic decomposition (fdepcad), real Nullstellensatz witness search based on Gröbner bases, exact branch-and-prune interval constraint propagation, and so on. Each of these techniques is embodied in a so-called *case manipulation function* (cmf) and produces a form of *proof trace*. Some of these proof traces give rise to easily checkable algebraic certificates which could be verified by a proof assistant with minimal library support for real algebra. Some, however, are at a much higher level. Proof traces for the fdepcad procedure, for instance, contain primitives such as real-root isolation ("p has exactly k real roots, and I is an isolating list of intervals for them"), signed subresultant computation ("the signed subresultant prs for p, q is R"), and liftable projection ("given $P \subset \mathbb{Q}[x_1, \ldots, x_n]$, a cad for $Q = Proj_n(P) \subset \mathbb{Q}[x_1, \ldots, x_{n-1}]$ can be lifted to a cad for P").

3 Discussion Questions

- What are some good approaches to replaying RAHD proofs in fully-expansive proof assistants? We'll talk about what the Coq team (F.Kirchner) has done for this.
- Imagine we knew proof assistant X could automatically replay RAHD proofs which used only cmfs $c_1, \ldots, c_k$. Would it be useful to be able to run RAHD in an "X-compatible mode," so that RAHD only searched for proofs which were currently automatically replayable in system X? Should we develop "X-compatible modes" for each X? What's the best way to go about this?
- Imagine an interactive "proof review system" in which users could navigate RAHD proofs and "verify by cosimulation" claims such as "the signed subresultant prs for p, q is R" by automatically running the relevant computations across many different computer algebra systems. To what extent would this contribute to trust? To what extent could a structured combination of algorithmic cosimulation and deductive verification become a robust form of social review for RAHD proofs (if it could at all)?