

How do we get inductive Invariants?

David Monniaux

CNRS, Verimag, Grenoble, France

Verifying the correctness of loop-free programs (or of general programs, up to bounded depth) is difficult: the state space explodes exponentially as the depth increases. Yet, the difficulty increases as we allow unboundedly many execution steps; proof approaches then generally rely on finding inductive invariants (properties shown to hold initially, then to remain true by induction).

Abstract interpretation attempts finding inductive invariants within a given domain, e.g. conjunctions of linear inequalities. The classical approach iterates a transformer until the property becomes inductive. In general, this approach may not terminate; thus termination is often enforced with a "widening" operator, which attempts at generalizing the iterates into an inductive property. Unfortunately, widening operators are brittle, with non-monotonic behaviors (supplying more information about a system may result in worse analysis outcomes!). Therefore, other approaches have been developed (policy iteration,...), which avoid this pitfall.

Finally, we shall discuss possible combinations of abstract interpretation and SMT-solving.