Acta Informatica (2019) 56:385–389
https://doi.org/10.1007/s00236-018-0324-y

ORIGINAL ARTICLE

On the decidability of the existence of polyhedral invariants in transition systems

David Monniaux[1,2]

Received: 13 September 2017 / Accepted: 9 May 2018 / Published online: 18 May 2018

Abstract Automated program verification often proceeds by exhibiting inductive invariants entailing the desired properties. For numerical properties, a classical class of invariants is convex polyhedra: solution sets of system of linear (in)equalities. Forty years of research on convex polyhedral invariants have focused, on the one hand, on identifying "easier" subclasses, on the other hand on heuristics for finding general convex polyhedra. These heuristics are however not guaranteed to find polyhedral inductive invariants when they exist. To our best knowledge, the existence of polyhedral inductive invariants has never been proved to be undecidable. In this article, we show that the existence of convex polyhedral invariants is undecidable, even if there is only one control state in addition to the "bad" one. The question is still open if one is not allowed any nonlinear constraint.

Mathematics Subject Classification 68Q17 · 68Q60 · 03D80 · 03D35 · 03D25 · 03B35

1 Introduction

Methods for proving that a program can never enter some undesirable states generally rely on exhibiting an *inductive invariant* I—a set of states containing the initial state(s) of the system, such that there exists no transition from I to outside of I—containing no undesirable states. If the program is safe, then the set of its reachable states is such an inductive invariant; this set is however in general very complex and approaches based on *abstract interpretation* [1] instead seek I within a restricted class called an *abstract domain*. In the case of programs

This work was partially supported by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreement No. 306595 "STATOR".

✉ David Monniaux
David.Monniaux@univ-grenoble-alpes.fr

[1] VERIMAG, Université Grenoble Alpes, CNRS, Grenoble INP (Institute of Engineering), 38000 Grenoble, France

[2] 700, Avenue Centrale, 38401 Saint Martin d'Hères, France

defined by a control flow graph (V, E), I is typically presented as a labeling of V. For instance, in *interval analysis* [1], each control location in is labeled either with the empty set, either with an interval for each numeric variable; the inductiveness of such a labeling may be checked by interval arithmetic.

Interval analysis however cannot express relationships between variables. In contrast, an example of *relational analysis* is that of *convex polyhedra* [2,4]: each control location is labeled with a set of linear (in)equalities over the program variables. An important question is: given a transition system, and an undesirable control location (for instance, expressing that arithmetic overflow or out-of-bound access has occurred), is there any inductive invariant defined by labeling each control location with a convex polyhedron such that the undesirable location is labeled with the empty set, and thus proved to be inaccessible?

To our best knowledge, all the literature on the inference of polyhedral invariants has focused on two issues: identifying "simpler" classes of convex polyhedra (the intervals form such a subclass), and heuristics for finding suitable (in)equalities. Cousot & Halbwachs' seminal work on polyhedral invariants [2,4] proposed three such heuristics: i) propagating polyhedra along the edges (computing their image by the transition on the edge), ii) computing the convex hull of all incoming polyhedra if there are several incoming edges, iii) "widening" polyhedra by discarding unstable (in)equalities, ensuring termination of the analysis. Further work included extracting potentially useful (in)equalities from the program, and less aggressive widening operators, in the hope of finding good invariants that coarser methods would have missed.

Yet none of these works considered showing that heuristics were needed in the first place, that is, showing that there is no general complete algorithm for finding inductive polyhedral invariants—such an algorithm would, given a transition system and an undesirable state, either find an inductive invariant proving that this state is inaccessible, either find that such an invariant does not exist.

In the case of some restricted classes of polyhedra, the existence or nonexistence of a suitable invariant is decidable; this is most notably the case if a "template", a finite catalog of possible normal vectors for the constraints, is fixed in advance. One may then study the complexity of this decision problem. If the program variables are real (as opposed to integer) and the transitions on the edges are expressed by a first-order linear arithmetic formula with an existential quantifier prefix, the problem is Σ_2^p-complete [3]. We shall see that, even when templates are not used, it is sufficient to bound the number of constraints per polyhedron to make the problem decidable. The issue is then what happens with an unbounded number of constraints.

In this article, we show that this problem is undecidable even for programs with one control state (plus one final "bad state"), as long as one can use at least one nonlinear (polynomial) condition.

2 Undecidability of the separating inductive polyhedral invariant existence problem

Definition 1 (Inductive separating invariant) Let $n \geq 0$, let $\mathcal{D} \subseteq \mathcal{P}(\mathbb{Q}^n)$ be a class of properties, and let $T \subseteq \mathcal{P}(\mathbb{Q}^n \times \mathbb{Q}^n)$ be a class of possible transition relations.

Let C be a set of control states (including c_i and c_f) and for all $j, k \in C$ let $\tau_{j,k} \subseteq \mathbb{Q}^n \times \mathbb{Q}^n$ be a transition relation defined by a formula in T; let $x_0 \in \mathbb{Q}^n$. Decide whether there is an inductive invariant $I \in \mathcal{D}^C$ such that $x_0 \in I(c_i)$ but $I(c_f) = \emptyset$—that is, I such that

$$\forall j, k \in C\ \forall x, x' \in \mathbb{Q}^n\ x \in I(j) \wedge (x, x') \in \tau_{j,k} \Rightarrow x' \in I(k) \tag{1}$$

Note that for $\mathcal{D} = \mathcal{P}(\mathbb{Q}^n)$, this problem is equivalent to the negation of reachability of c_f from c_i, thus, if $n \geq 2$ and $\mathcal{T}$ includes first order linear arithmetic, this problem is undecidable, by simulation of a two-counter machine.

Remark 1 Consider the above definition with a real state $x \in \mathbb{R}^n$. If $\mathcal{D}$ is the set of convex polyhedra over $\mathbb{R}^n$ with at most k constraints, and the transitions contain linear arithmetic updates and polynomial guards, the inductive separation problem is decidable.

Proof The stability condition (1) is then a formula in the theory of real closed fields, with free variables the coefficients of the k polyhedral constraints. We thus reduce the problem of the existence of the invariant to that of satisfiability of such a formula, which is decidable. □

Theorem 1 *If $\mathcal{D}$ is the set of convex polyhedra over $\mathbb{Q}^n$, and the transitions contain (deterministic) linear arithmetic assignments and polynomial guards, the inductive separation problem is undecidable.*

Proof We reduce the halting problem for a deterministic machine over m registers and s states, with linear updates and linear (or polynomial) arithmetic constraints on the transitions, to the problem of existence of a convex polyhedral inductive separating invariant for a machine with $m + 2$ registers and $s + 1$ states, again with linear updates and polynomial guards on the transitions.

The target machine is obtained from the source as follows:

- the control states are the same, to which we add a special "bad state" σ_b
- the first m registers encode the registers of the source machine
- another register, called t, is a time counter: initialized to 0, it is incremented at every step
- another register, called y, is initialized to 0 and updated as $y := y + t + 1$
- the transitions are the same as those of the source machine, conjoined with a $y = (t^2+t)/2$ guard ($y \leq (t^2 + t)/2$ also works), to which we add transitions from all states to σ_b with the guard $y < (t^2 + t)/2$.

If the source machine terminates, then there exists a bounded polyhedral inductive invariant Let $(\sigma_i, \mathbf{x}_i)_{1 \leq i \leq n}$ be the run of the source machine, when σ_i is the control state and $\mathbf{x}_i$ the vector of registers at time i. The run of the target machine is thus $(\sigma_i, \mathbf{x}_i, i, (i + i^2)/2)_{1 \leq i \leq n}$.

For each control state σ, consider the bounded polyhedron obtained as the convex hull of those $(\sigma, \mathbf{x}_i, i, (i + i^2)/2)_{1 \leq i \leq n}$ from the run (this polyhedron is thus empty if and only if this control state is unreachable). This polyhedron lies above the $y \geq (t^2 + t)/2$ parabola; its points are partitioned into, on the one hand, the $(\sigma_i, \mathbf{x}_i, i, (i + i^2)/2)_{1 \leq i \leq n}$, on the other hand, points satisfying $y > (t^2 + t)/2$ and thus excluded from the transitions by the $y = (t^2 + t)/2$ guard.

The collection of these polyhedra thus forms an inductive invariant, with σ_b labeled with the empty polyhedron.

If there exists a polyhedral separating invariant, then the source machine terminates If any point in the polyhedra lies below $y = (t^2 + t)/2$, then σ_b is reachable and thus the invariant is not separating; and any point on $y = (t^2 + t)/2$ must be a vertex.

Suppose an infinite run $(\sigma_i, \mathbf{x}_i, i, (i + i^2)/2)_{1 \leq i}$, then there exists at least one of the control states σ that repeats infinitely. The corresponding $(\sigma_i, \mathbf{x}_i, i, (i + i^2)/2)$ are vertices of the polyhedron associated to σ, but there is an infinity of them, which is absurd. □

The same theorem and proof are valid for the set of bounded convex polyhedra over $\mathbb{Q}^n$, the sets of bounded and unbounded convex polyhedra over $\mathbb{Z}^n$.

3 Reduction to two control states

We consider a transition system operating over $d \geq 1$ integer (resp. rational) variables and $N \geq 1$ control points. 0 shall be the initial control point and $N - 1$ is the final "bad state".

The state of the system consists in $\langle q, x_1, \ldots, x_d \rangle$ where $0 \leq q < N$ is a control state and $x_1, \ldots, x_d$ are integer (resp. rational) variables. The initial state is $\sigma_0 \overset{\triangle}{=} \langle 0, 0, \ldots, 0 \rangle$. The set of states Σ is thus $\{0, \ldots, N-1\} \times \mathbb{Z}^d$ (resp. $\{0, \ldots, N-1\} \times \mathbb{Q}^d$). In the τ and B formula, we request, without loss of generality, the only atomic propositions involving q to be of the form $q = k$ where $0 \leq k < N$ is a constant.

Theorem 2 *The inductive separating polyhedral invariant problem for N control states in dimension d reduces to the inductive separating polyhedral invariant problem for 2 control states in dimension $d + (N - 1)$.*

Proof In the new system of transitions, a state consists in $\langle x_1, \ldots, x_d, y_1, \ldots, y_{N-1} \rangle$. The variables $x_1, \ldots, x_d$ encode the registers of the original system. The variables $y_1, \ldots, y_{N-1}$ indicate which control state q of the original system is simulated by encoding it into a vertex of a simplex: $q = 0$ is encoded as $\langle 0, \ldots, 0 \rangle$, and $q > 0$ is encoded into $e_q = \langle \underbrace{0, \ldots, 0}_{q-1 \text{ times}}, 1, 0, \ldots, 0 \rangle$.

The transition relation τ' the new system is obtained by replacing $q = k$ in each formula by $\langle y_1, \ldots, y_{N-1} \rangle = e_q$ where e_q is the encoding of control state q.

The target system simulates concretely the source system, and conversely; the concrete reachability problems are equivalent.

Let us now see that any polyhedral inductive invariant within the source system (one polyhedron per control location) can be translated into a polyhedral invariant within the target system (one polyhedron, for the single control location), and the converse.

A polyhedral inductive invariant within the source system consists in one polyhedron P_q per control location q; it is translated into a polyhedron obtained as the convex hull of $\bigcup_q P_q \times \{e_q\}$. Conversely, a polyhedral invariant P in the target system is translated into the polyhedral invariant defined as $P_q = \{\langle x_1, \ldots, x_d \rangle \mid (\langle x_1, \ldots, x_d \rangle, e_q) \in P\}$. □

4 Perspectives

Most studies of the computability and complexity issues in hardware and software verification have focused on the (concrete) reachability problem for various classes of transition systems. Here, we instead consider the problem that abstract interpreters (or other classes of tools, for instance those based on counterexample-guided abstraction refinement) solve heuristically: finding an invariant within a given class; or, to make it a decision problem, deciding the existence of such an invariant. [1]

[1] For most abstract domains, it is possible to find a suitable invariant if one is known to exist, by enumerating all candidate invariants and stopping when an inductive separating one is found. This justifies the use of the existence decision problem for the purposes of theoretical computability studies.

We started this work hoping to vindicate forty years of research on heuristics by showing that the existence of polyhedral inductive separating invariants in a system with transitions in linear arithmetic (integer or rational) is undecidable; we had to settle for a weaker result (involving nonlinear arithmetic). The original question is still open; we have discussed it with experts, to no avail.

References

1. Cousot, P., Cousot, R.: Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Graham, R.M., Harrison, M.A., Sethi, R. (eds) Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, Jan 1977, ACM, pp 238–252 (1977). https://doi.org/10.1145/512950.512973
2. Cousot, P., Halbwachs, N.: Automatic discovery of linear restraints among variables of a program. In: Aho, A.V., Zilles, S.N., Szymanski, T.G. (eds.) POPL, pp. 84–96. ACM Press, New York (1978). https://doi.org/10.1145/512760.512770
3. Gawlitza, T., Monniaux, D.: Invariant generation through strategy iteration in succinctly represented control flow graphs. Logical Methods in Computer Science (2012). https://doi.org/10.2168/LMCS-8(3:29)2012
4. Halbwachs, N.: Détermination automatique de relations linéaires vérifiées par les variables d'un programme. Ph.D. thesis, Université Scientifique et Médicale de Grenoble and Institut National Polytechnique de Grenoble (1979). https://tel.archives-ouvertes.fr/tel-00288805