Interprocedural Algebraic Invariants

James Worrell

Department of Computer Science
Oxford University

IFIP WG 2.2 Aachen, 2025

Invariants: a Tale as Old as Time

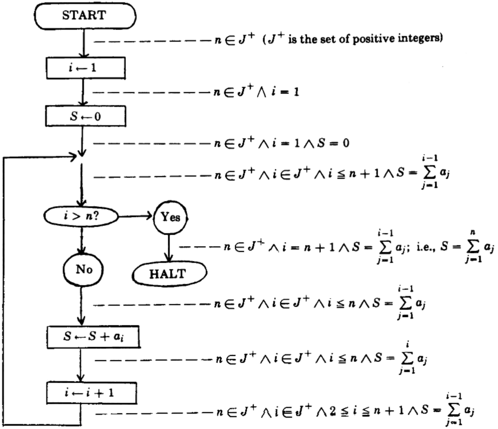
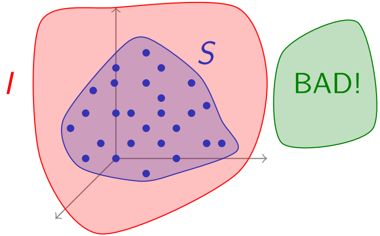


FIGURE 1. Flowchart of program to compute $S = \sum_{j=1}^{n} a_j$ $(n \geqq 0)$

Robert W. Floyd, *Assigning Meanings to Programs*, 1967

Invariants and Verification



The classical approach to the verification of temporal safety properties of programs requires the construction of inductive invariants [...]. **Automation of this construction is the main challenge in program verification.**

D. Beyer, T. Henzinger, R. Majumdar, and A. Rybalchenko
Invariant Synthesis for Combined Theories, 2007

Polyhedral and Algebraic Invariants

Algorithm 1: Loop Program with Invariant

$x \leftarrow 0$;
$y \leftarrow 0$;
while $x < 100$ **do**
    $x \leftarrow x + 1$;
    $y \leftarrow y + x$;

Invariant: $2y - x^2 - x = 0 \quad \wedge \quad 0 \leq x \leq 100$

Application to Transducers @ FOCS'15

Equivalence of Deterministic Top-Down Tree-to-String Transducers Is Decidable

HELMUT SEIDL, Technical University of Munich
SEBASTIAN MANETH, Universität of Bremen
GREGOR KEMPER, Technical University of Munich

> *"[...]we introduce polynomial transducers and prove that for these, equivalence can be certified by means of an inductive polynomial invariant. This allows us to construct two semi-algorithms, one searching for an invariant and the other for a witness of non-equivalence [...]"*

Application to Quantum Automata

DECIDABLE AND UNDECIDABLE PROBLEMS ABOUT QUANTUM AUTOMATA*

VINCENT D. BLONDEL†, EMMANUEL JEANDEL‡, PASCAL KOIRAN‡, AND NATACHA PORTIER‡

Abstract. We study the following decision problem: is the language recognized by a quantum finite automaton empty or nonempty? We prove that this problem is decidable or undecidable depending on whether recognition is defined by strict or nonstrict thresholds. This result is in contrast with the corresponding situation for probabilistic finite automata, for which it is known that strict and nonstrict thresholds both lead to undecidable problems.

Theorem (Blondel, Jeandel, Koiran, Portier 2005)

The strict threshold problem is decidable for quantum automata.

Invariants for Affine Programs @ ICALP'04

A Note on Karr's Algorithm

Markus Müller-Olm[1*] and Helmut Seidl[2]

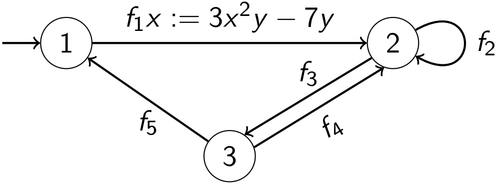
Abstract. We give a simple formulation of Karr's algorithm for computing all affine relationships in affine programs. This simplified algorithm runs in time $\mathcal{O}(nk^3)$ where n is the program size and k is the number of program variables assuming unit cost for arithmetic operations. This improves upon the original formulation by a factor of k. Moreover, our re-formulation avoids exponential growth of the lengths of intermediately occurring numbers (in binary representation) and uses less complicated elementary operations. We also describe a generalization that determines all polynomial relations up to degree d in time $\mathcal{O}(nk^{3d})$.

Theorem

There is an algorithm which computes, for any given **affine program**, *all its polynomial inductive invariants up to any fixed degree d.*

Polynomial and Affine Programs

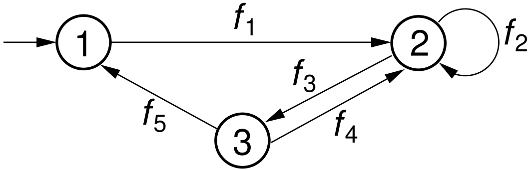
Polynomial Programs (Muller-Olm and Seidl 2004)

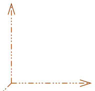


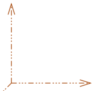
- Nondeterministic branching (no guards)
- All assignments are affine (or polynomial)
- Compute **all valid polynomial equations** at each location
- Represents the **Zariski closure** of the reachable set at each location

Polynomial Invariants: Geometric Picture

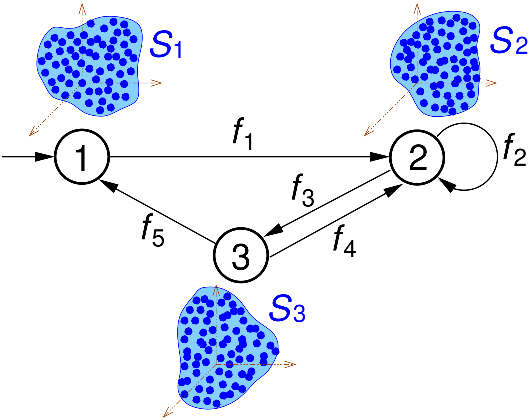
x, y, z range over $\mathbb{Q}$

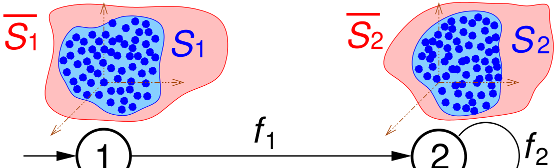






Computing Strongest **Inductive** Polynomial Invariants





Finding all polynomial invariants

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Information Processing Letters 91 (2004) 233–244

Information Processing Letters
www.elsevier.com/locate/ipl

Computing polynomial program invariants

Markus Müller-Olm [a,*,1], Helmut Seidl [b]

[a] FernUniversität Hagen, LG Praktische Informatik 5, 58084 Hagen, Germany
[b] TU München, Informatik, I2, 85748 München, Germany

Received 16 October 2003; received in revised form 20 April 2004

Available online 19 June 2004

It is a challenging open problem whether or not the set of *all* valid polynomial relations can be computed not just the ones of some given form. It is not

Strongest Polynomial Invariants @ LICS'18

Theorem (Hrushovski, Ouaknine, Pouly, W. 18)

There is an algorithm which computes the strongest polynomial inductive invariant of a given affine program.

- Algorithm computes for each location the set of **all polynomial relations** among program variables that hold whenever control reaches that location
- We represent this set of relations using a **finite basis** of polynomial equalities
- Dually, the algorithm computes for each location the **smallest algebraic set** containing the set of reachable states

Affine Programs with Recursive Procedures

```
procedure Q()
  begin
    if (*) then
      x := Ax ; call Q() ; x := Bx ; call Q()
    else
      skip
    endif
  end
```

Compute $\overline{\varphi(L)}$ for Dyck language $L \subseteq \{a, b\}^*$, where $\varphi(a) = A$ and $\varphi(b) = B$.

Interprocedural Equational Invariants @ POPL'04

Precise Interprocedural Analysis through Linear Algebra

Markus Müller-Olm *
FernUniversität Hagen, LG Praktische Informatik 5
58084 Hagen, Germany
mmo@ls5.cs.uni-dortmund.de

Helmut Seidl
TU München, Lehrstuhl für Informatik II
80333 München, Germany
seidl@informatik.tu-muenchen.de

[...] we describe analyses that determine identities valid among program variables at each program point. Our analyses interpret assignment statements with affine expressions on the right hand side and ignore conditions at branches. Under this abstraction, the analysis computes all polynomial relations of bounded degree precisely.

Interprocedural Invariant Synthesis

Theorem (Ait El Manssour, Naraghi, Shirmohammadi, W. 25)
Given a morphism $\varphi : \Sigma^* \to M_d(\mathbb{Q})$, we can compute $\overline{\varphi(L)}$ if either L is a one-counter language or L is context-free and φ takes values in invertible matrices.

Proof. Analog of Simon's factorisation forest theorem for matrix semigroups.

Theorem (Ait El Manssour, Naraghi, Shirmohammadi, W. 25)
There is no algorithm to compute $\overline{\varphi(L)}$ for L the language of an indexed grammar.

Can we compute $\overline{\varphi(L)}$ for a context-free language L and general φ?

The Monniaux Problem

P. Cousot

N. Halbwachs

D. Monniaux

"*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.*"

– David Monniaux, 2019

Interprocedurally Analysing Linear Inequality Relations

Helmut Seidl, Andrea Flexeder, and Michael Petter

Technische Universität München, Boltzmannstrasse 3, 85748 Garching, Germany
{seidl,flexeder,petter}@cs.tum.edu
http://www2.cs.tum.edu/~{seidl,flexeder,petter}

[. . .] we present an alternative approach to interprocedurally inferring linear inequality relations. We propose an abstraction of the effects of procedures through convex sets of transition matrices. In the absence of conditional branching, this abstraction can be characterised precisely by means of the least solution of a constraint system [. . .]

Undecidability

Theorem (Hrushovski, Ouaknine, Pouly, W. 23)

There is no algorithm that computes the Zariski closure of the reachable set of a polynomial program.

Theorem (Monniaux 19)

There is no algorithm for determining the existence of convex polyhedral separating invariants for polynomial programs.

Theorem (Fijalkow *et al.* 25)

There is no algorithm for certifying non-reachability in affine programs by (non-convex) polyhedral invariants.

Outstanding Problems

- Is there an algorithm to compute all algebraic invariants for affine programs with recursion?
- Is there a procedure for determining existence of convex polyhedral invariants for affine programs (with and without recursion)?
- Is it decidable whether a finite set of matrices preserves some bounded convex polyhedron?