Polynomial Invariants for Affine Programs

Ehud Hrushovski, Joël Ouaknine, Amaury Pouly, James Worrell

Max Planck Institute for Software Systems &
Department of Computer Science, Oxford University &
Mathematical Institute, Oxford University

Does this program halt?

Affine program

$x := 2^{-10}$
$y := 1$
while $y \geqslant x$ do

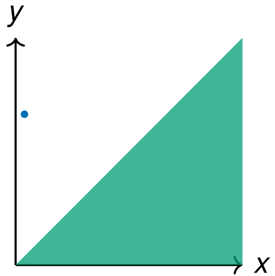
$$\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} 2 & 0 \\ \frac{7}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Does this program halt?

Affine program

$x := 2^{-10}$
$y := 1$
while $y \geqslant x$ do

$$\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} 2 & 0 \\ \frac{7}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

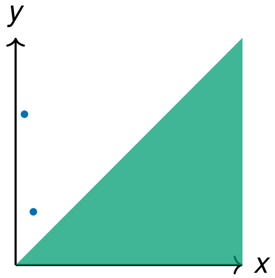


Does this program halt?

Affine program

$x := 2^{-10}$
$y := 1$
while $y \geqslant x$ do

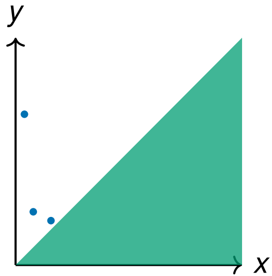
$$\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} 2 & 0 \\ \frac{7}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



Does this program halt?

Affine program

$x := 2^{-10}$
$y := 1$
while $y \geqslant x$ do
$$\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} 2 & 0 \\ \frac{7}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



Does this program halt?

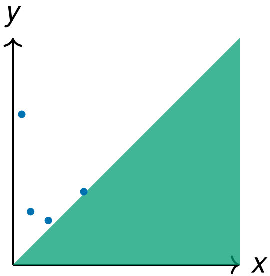
Affine program

$x := 2^{-10}$

$y := 1$

while $y \geqslant x$ do

$$\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} 2 & 0 \\ \frac{7}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

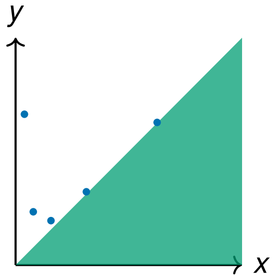


Does this program halt?

Affine program

$x := 2^{-10}$
$y := 1$
while $y \geqslant x$ do

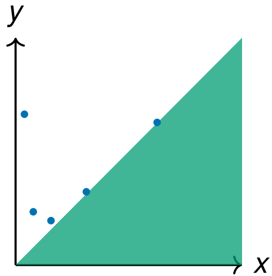
$$\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} 2 & 0 \\ \frac{7}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



Does this program halt?

Affine program

$x := 2^{-10}$
$y := 1$
while $y \geqslant x$ do
$$\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} 2 & 0 \\ \frac{7}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



Certificate of non-termination:

$$x^2y - x^3 = \frac{1023}{1073741824} \tag{1}$$

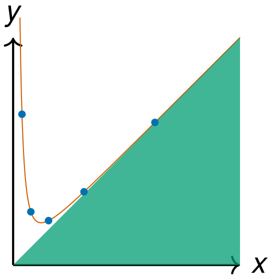
Does this program halt?

Affine program

$x := 2^{-10}$
$y := 1$
while $y \geqslant x$ do
$$\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} 2 & 0 \\ \frac{7}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Certificate of non-termination:

$$x^2y - x^3 = \frac{1023}{1073741824} \qquad (1)$$



- (1) is an invariant: it holds at every step

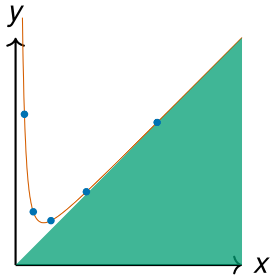
Does this program halt?

Affine program

$x := 2^{-10}$
$y := 1$
while $y \geqslant x$ do
$$\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} 2 & 0 \\ \frac{7}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Certificate of non-termination:

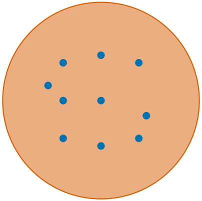
$$x^2 y - x^3 = \frac{1023}{1073741824} \tag{1}$$



- (1) is an invariant: it holds at every step
- (1) implies the guard is true

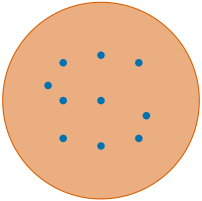
Invariants

invariant = overapproximation of the reachable states

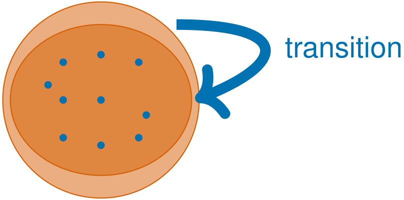


Invariants

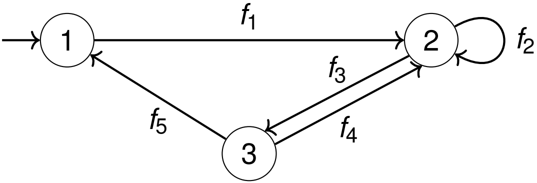
invariant = overapproximation of the reachable states



inductive invariant = invariant preserved by the transition relation

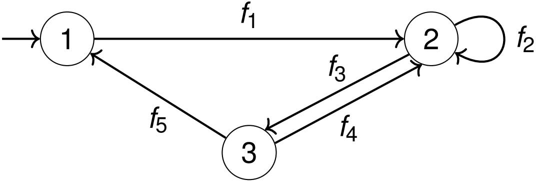


Inductive invariants: example



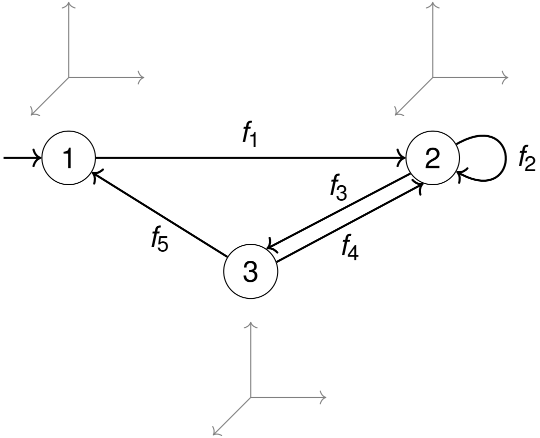
Inductive invariants: example

x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



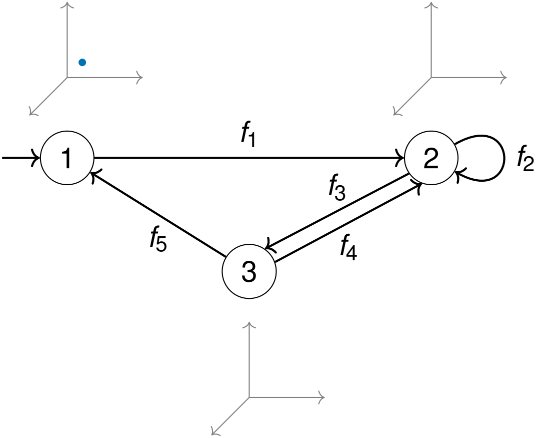
Inductive invariants: example

x, y, z range over $\mathbb{Q}$ $\qquad\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



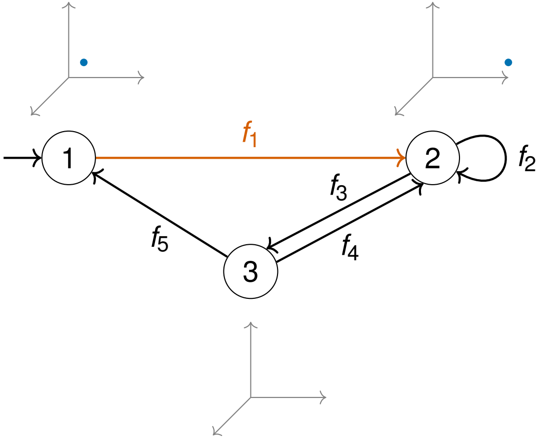
Inductive invariants: example

x, y, z range over $\mathbb{Q}$ $\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



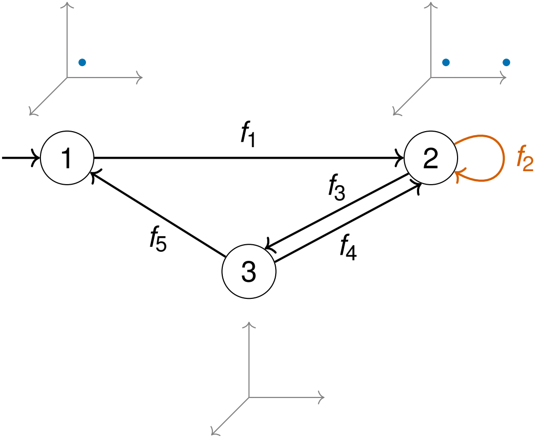
Inductive invariants: example

x, y, z range over $\mathbb{Q}$ $\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



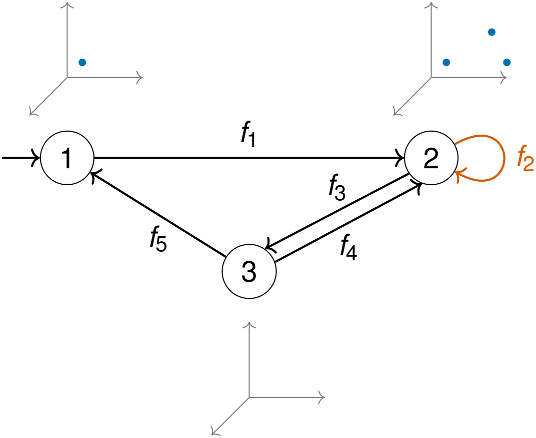
Inductive invariants: example

x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



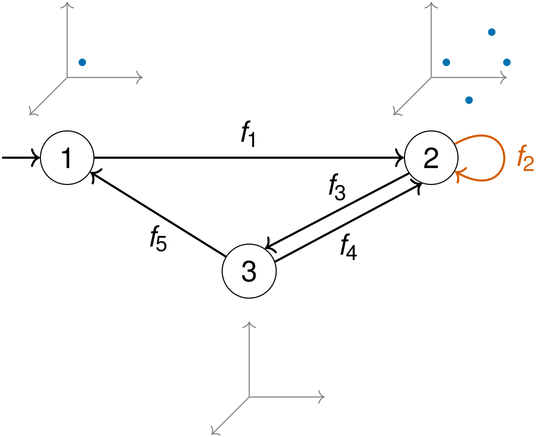
Inductive invariants: example

x, y, z range over $\mathbb{Q}$ $\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



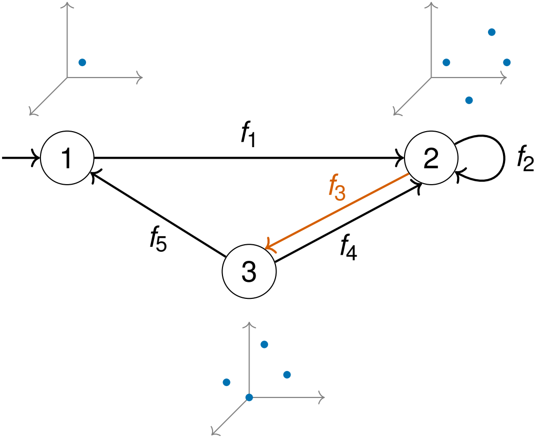
Inductive invariants: example

x, y, z range over $\mathbb{Q}$ $\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



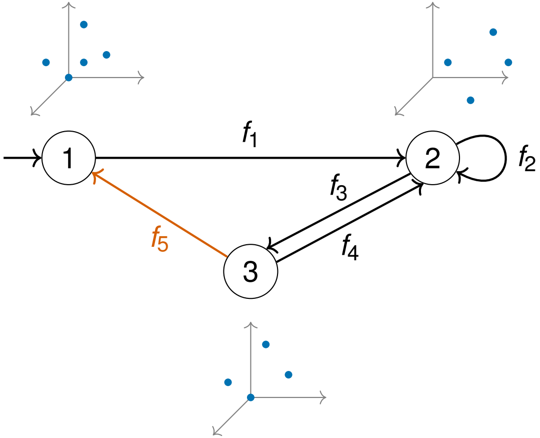
Inductive invariants: example

x, y, z range over $\mathbb{Q}$ $\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



Inductive invariants: example

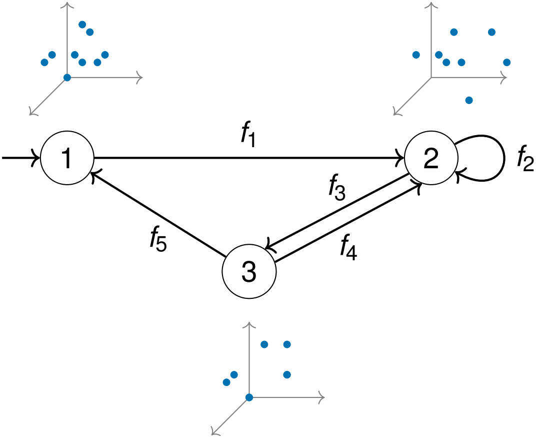
x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



Inductive invariants: example

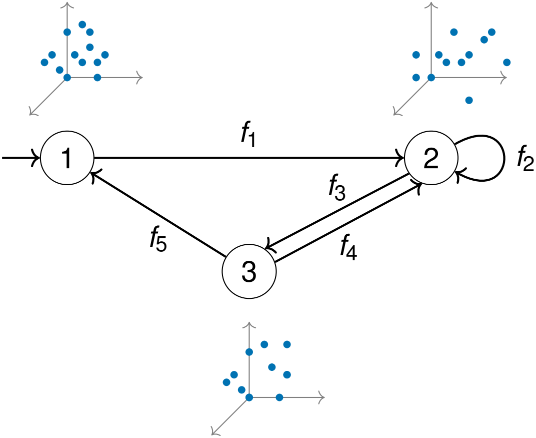
x, y, z range over $\mathbb{Q}$

$f_i : \mathbb{R}^3 \to \mathbb{R}^3$



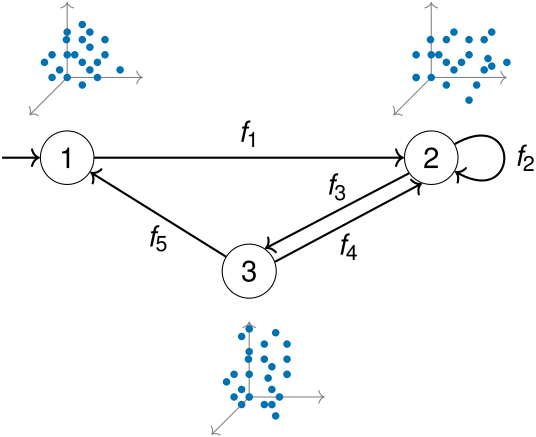
Inductive invariants: example

x, y, z range over $\mathbb{Q}$ $\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



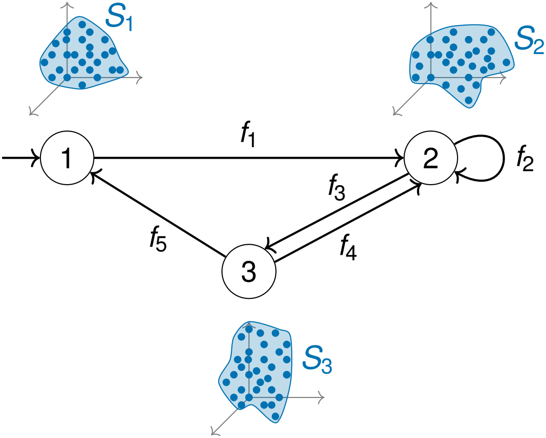
Inductive invariants: example

x, y, z range over $\mathbb{Q}$ $\quad\quad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



Inductive invariants: example

x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$

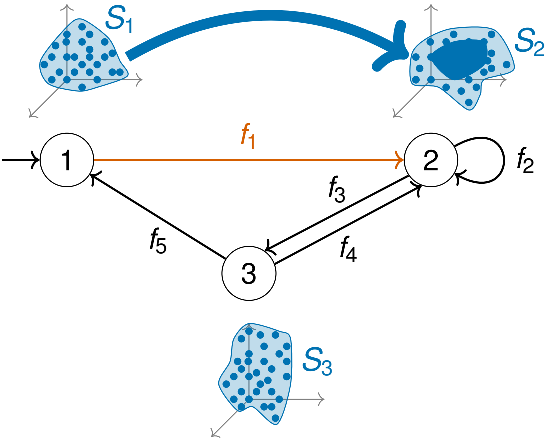


S_1, S_2, S_3 is an invariant

Inductive invariants: example

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

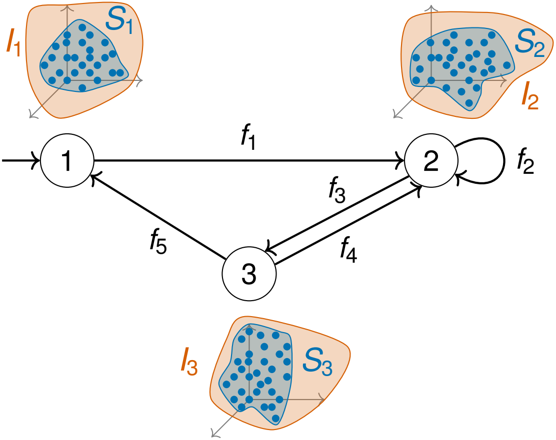
$f_i : \mathbb{R}^3 \to \mathbb{R}^3$



S_1, S_2, S_3 is an inductive invariant

Inductive invariants: example

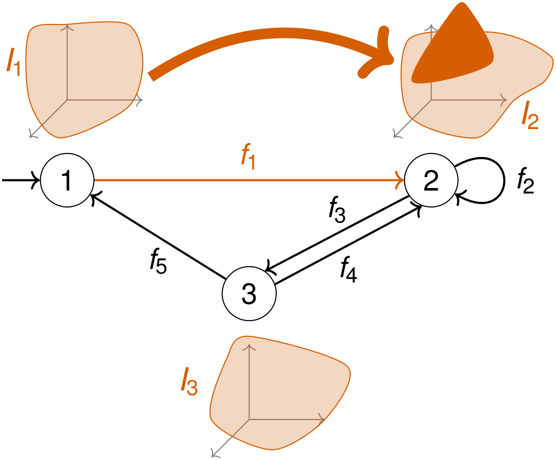
x, y, z range over $\mathbb{Q}$ $\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



I_1, I_2, I_3 is an invariant

Inductive invariants: example

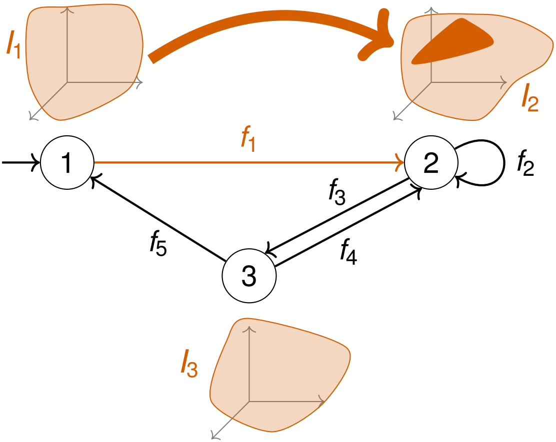
x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



I_1, I_2, I_3 is **NOT** an inductive invariant

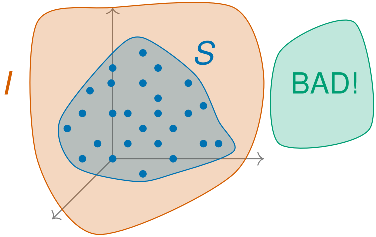
Inductive invariants: example

x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



I_1, I_2, I_3 is an inductive invariant

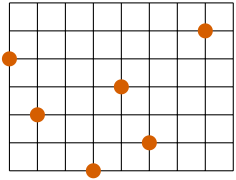
Why Invariants?



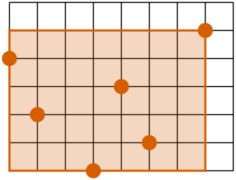
*The classical approach to the verification of temporal safety properties of programs requires the construction of **inductive invariants** [...]. **Automation of this construction is the main challenge in program verification.***

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

Which invariants?

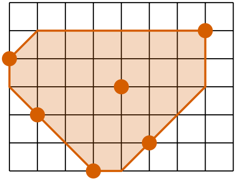


Which invariants?



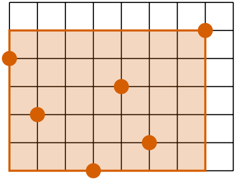
Intervals

Which invariants?



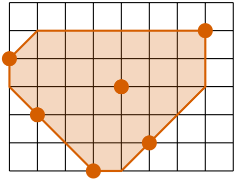
Octagons

$\leq$



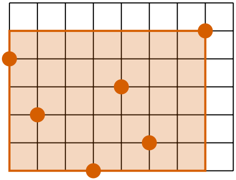
Intervals

Which invariants?

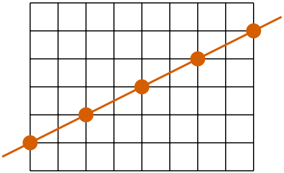


Octagons

$\leq$

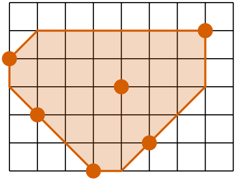


Intervals



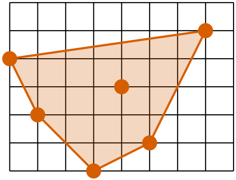
Affine/linear sets

Which invariants?



Octagons

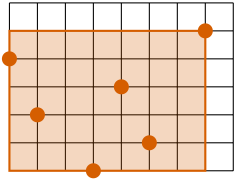
$\leqslant$



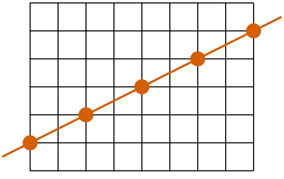
Polyhedrons

$\leqslant$

$\leqslant$

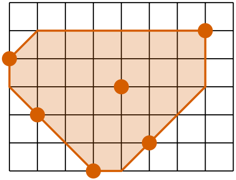


Intervals



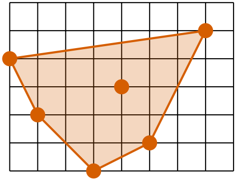
Affine/linear sets

Which invariants?



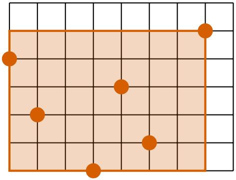
Octagons

$\leqslant$

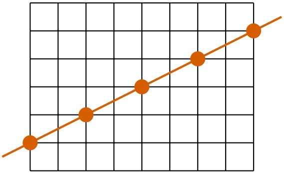


Polyhedrons

$\veeeq$ $\veeeq$

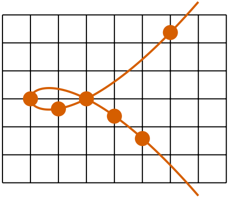


Intervals



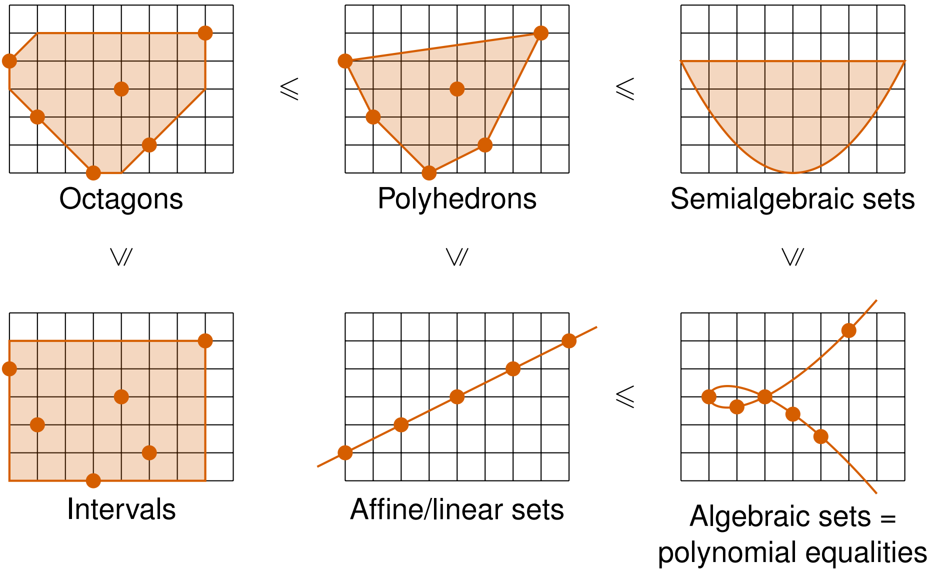
Affine/linear sets

$\leqslant$

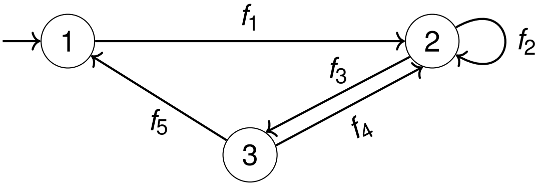


Algebraic sets = polynomial equalities

Which invariants?

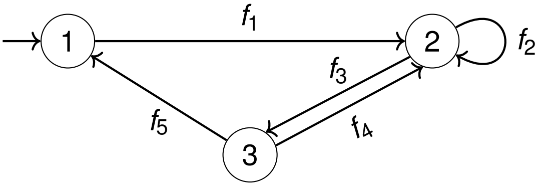


Affine programs



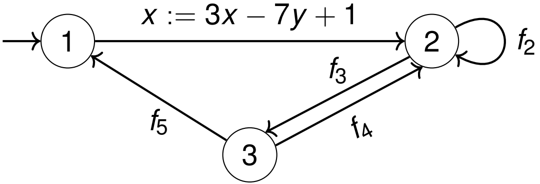
Affine programs

- Nondeterministic branching (no guards)



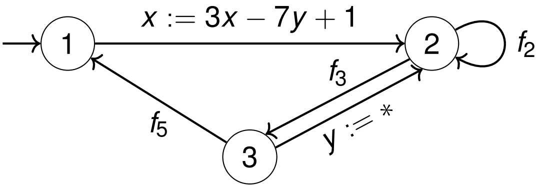
Affine programs

- Nondeterministic branching (no guards)
- All assignments are affine



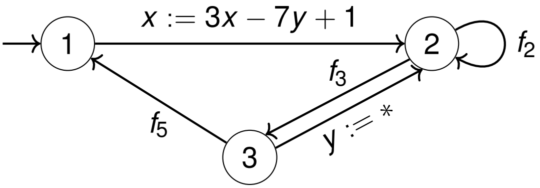
Affine programs

- Nondeterministic branching (no guards)
- All assignments are affine
- Allow nondeterministic assignments ($x := *$)



Affine programs

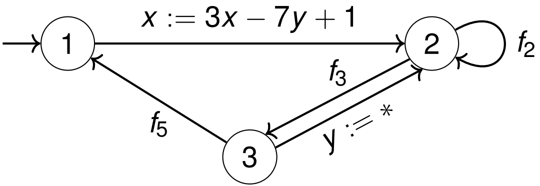
- Nondeterministic branching (no guards)
- All assignments are affine
- Allow nondeterministic assignments ($x := *$)



- Can overapproximate complex programs

Affine programs

- Nondeterministic branching (no guards)
- All assignments are affine
- Allow nondeterministic assignments ($x := *$)



- Can overapproximate complex programs
- Covers existing formalisms:
  probabilistic, quantum, quantitative automata

Karr's Algorithm

Affine Relationships Among Variables of a Program*

Michael Karr

Received May 8, 1974

Summary. Several optimizations of programs can be performed when in certain regions of a program equality relationships hold between a linear combination of the variables of the program and a constant. This paper presents a practical approach to detecting these relationships by considering the problem from the viewpoint of linear algebra. Key to the practicality of this approach is an algorithm for the calculation of the "sum" of linear subspaces.

Theorem (Karr 76)

There is an algorithm which computes, for any given affine program over $\mathbb{Q}$, its strongest affine inductive invariant.

Randomized Karr's Algorithm @ POPL 2003

Discovering Affine Equalities Using Random Interpretation

Sumit Gulwani George C. Necula
University of California, Berkeley
{gulwani,necula}@cs.berkeley.edu

ABSTRACT

We present a new polynomial-time randomized algorithm for discovering affine equalities involving variables in a program.

Keywords

Affine Relationships, Linear Equalities, Random Interpretation, Randomized Algorithm

Some polynomial invariants

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 (ICALP 2004)

*There is an algorithm which computes, for any given affine program over $\mathbb{Q}$, all its polynomial inductive invariants up to any **fixed degree** d.*

A challenge: finding all polynomial invariants

ELSEVIER

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

A challenge: finding all polynomial invariants

ELSEVIER

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]

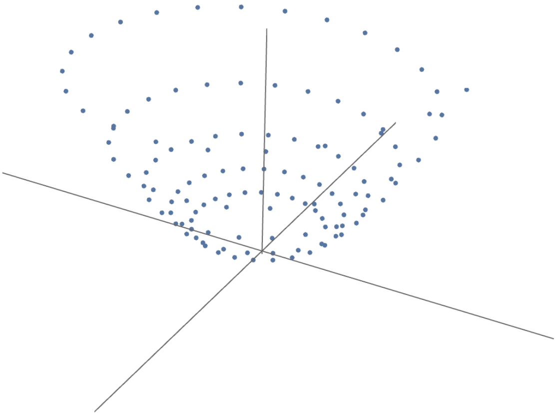
[a] FernUniversität Hagen, LG Praktische Informatik 5, 58084 Hagen, Germany
[b] TU München, Informatik, I2, 85748 München, Germany

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

Available online 19 June 2004

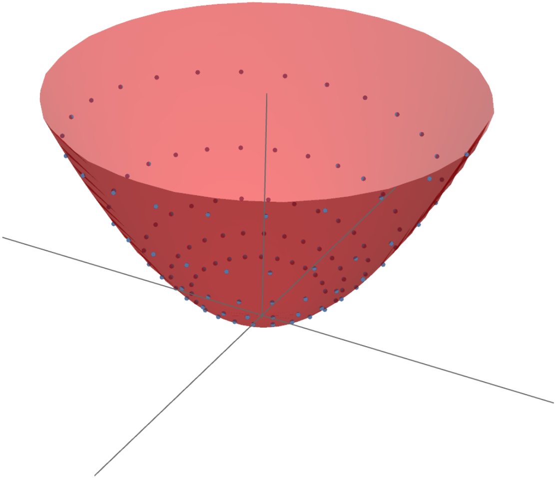
It is a challenging open problem whether or not the set of *all* valid polynomial relations can be computed not just the ones of some given form. It is not

Why fixed degree is not enough



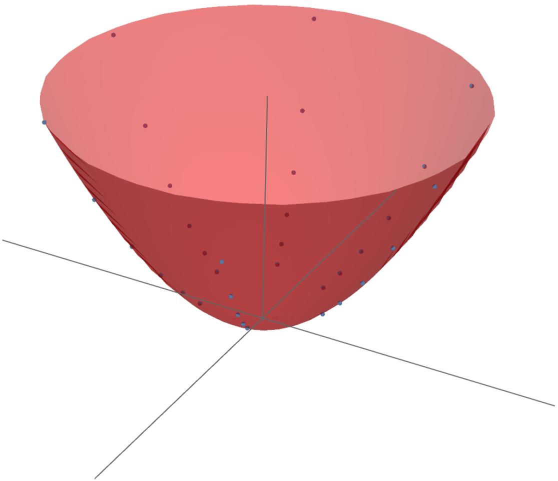
Why fixed degree is not enough

- Paraboloid $z = x^2 + y^2$



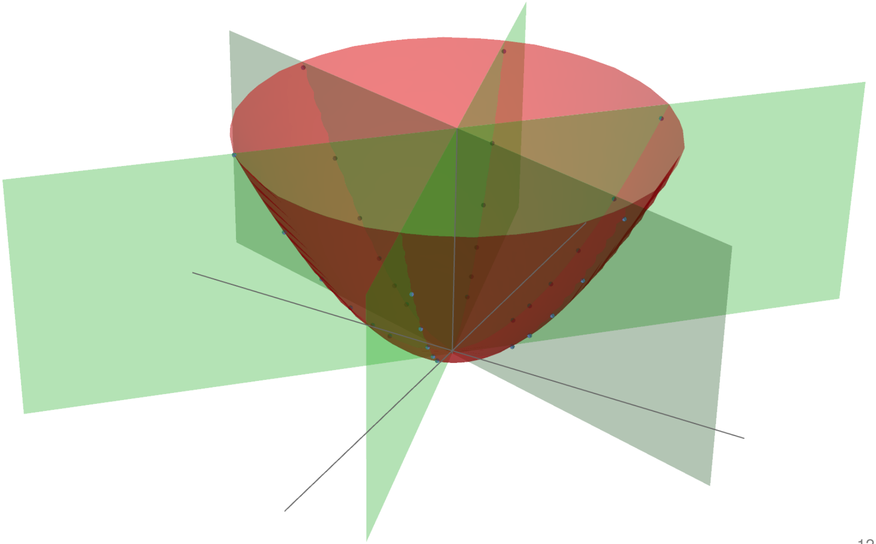
Why fixed degree is not enough

- Paraboloid $z = x^2 + y^2$



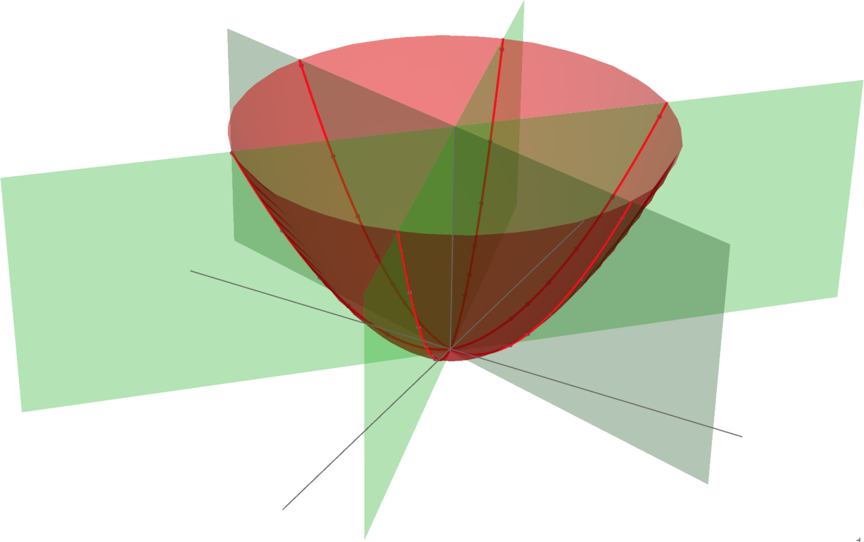
Why fixed degree is not enough

- Paraboloid $z = x^2 + y^2$
- Union of 3 hyperplanes $(x - y)(10y + x)(y + 10x) = 0$



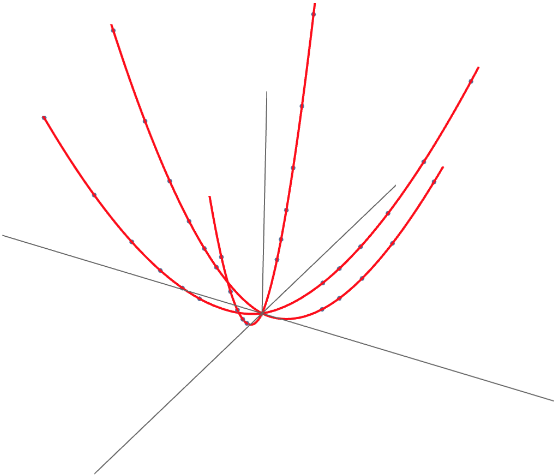
Why fixed degree is not enough

- Paraboloid $\quad z = x^2 + y^2$
- Union of 3 hyperplanes $\quad (x - y)(10y + x)(y + 10x) = 0$



Why fixed degree is not enough

- Paraboloid $$z = x^2 + y^2$$
- Union of 3 hyperplanes $$(x - y)(10y + x)(y + 10x) = 0$$



Main result

Theorem

There is an algorithm which computes, for any given affine program over $\mathbb{Q}$, its strongest polynomial inductive invariant.

Main result

Theorem

There is an algorithm which computes, for any given affine program over $\mathbb{Q}$, its strongest polynomial inductive invariant.

- strongest polynomial invariant $\iff$ smallest algebraic set

Main result

Theorem

There is an algorithm which computes, for any given affine program over $\mathbb{Q}$, its strongest polynomial inductive invariant.

- strongest polynomial invariant $\iff$ smallest algebraic set
  - algebraic sets = finite $\bigcup$ and $\bigcap$ of polynomial equalities

Main result

Theorem

There is an algorithm which computes, for any given affine program over $\mathbb{Q}$, its strongest polynomial inductive invariant.

- strongest polynomial invariant $\iff$ smallest algebraic set
  - algebraic sets = finite $\bigcup$ and $\bigcap$ of polynomial equalities
- Thus our algorithm computes all polynomial relations that always hold among program variables at each program location, in all possible executions of the program

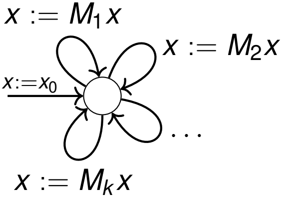
Main result

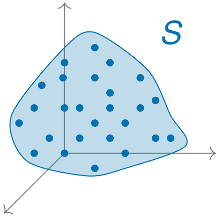
Theorem

There is an algorithm which computes, for any given affine program over $\mathbb{Q}$, its strongest polynomial inductive invariant.

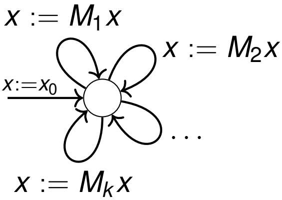
- strongest polynomial invariant $\iff$ smallest algebraic set
  - algebraic sets = finite $\bigcup$ and $\bigcap$ of polynomial equalities
- Thus our algorithm computes all polynomial relations that always hold among program variables at each program location, in all possible executions of the program
- We can represent this (usually infinite) set of relations using a finite basis of polynomial equalities

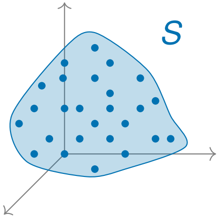
At the edge of decidability





At the edge of decidability



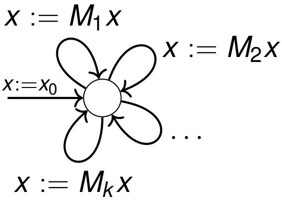


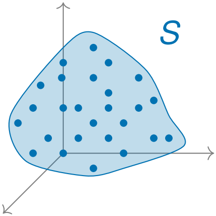
Theorem (Markov 1947*)

There is a fixed set of 6×6 *integer matrices* $M_1, \ldots, M_k$ *such that the reachability problem "y is reachable from x_0?" is undecidable.*

*Original theorems about semigroups, reformulated with affine programs.

At the edge of decidability





Theorem (Markov 1947*)

There is a fixed set of 6×6 *integer matrices* $M_1, \ldots, M_k$ *such that the reachability problem "y is reachable from x_0?" is undecidable.*

Theorem (Paterson 1970*)

The mortality problem "0 is reachable from x_0 with $M_1, \ldots, M_k$?" is undecidable for 3×3 *matrices.*

*Original theorems about semigroups, reformulated with affine programs.

Tools

- Algebraic geometry
- Number theory
- Group theory

Tools

- Algebraic geometry
- Number theory
- Group theory

Quantum automata and algebraic groups

Harm Derksen[a], Emmanuel Jeandel[b], Pascal Koiran[b,*]

[a]*Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, United States*
[b]*Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon, 69364, France*

Received 15 September 2003; accepted 1 November 2004

Theorem (Derksen, Jeandel and Koiran, 2004)

There is an algorithm which computes, for any given affine program over $\mathbb{Q}$ using only invertible transformations, its strongest polynomial inductive invariant.

Main contribution

Theorem

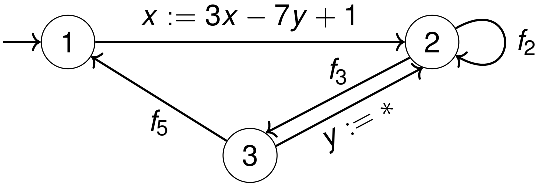
Given a finite set of rational square matrices of the same dimension, we can compute the Zariski closure of the semigroup that they generate.

Corollary

Given an affine program, we can compute for each location the ideal of all polynomial relations that hold at that location.

Summary

- invariant = overapproximation of reachable states
- invariants allow verification of safety properties
- affine program:
  - nondeterministic branching, no guards, affine assignments



Theorem

There is an algorithm which computes, for any given affine program over $\mathbb{Q}$, its strongest polynomial inductive invariant.