Linear Dynamical Systems

Invariant Synthesis

Amaury Pouly

Does this program halt?

Affine program

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

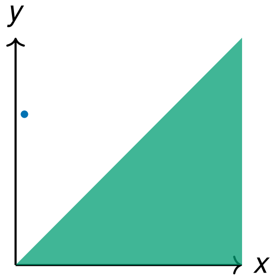
$$\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} 2 & 0 \\ \frac{7}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Does this program halt?

Affine program

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

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

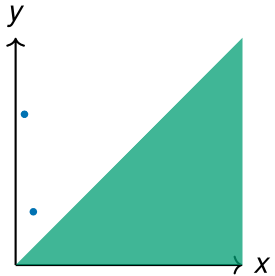


Does this program halt?

Affine program

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

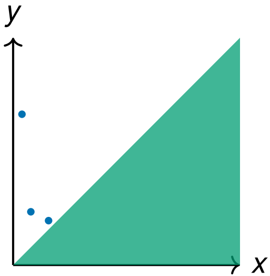
$$\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} 2 & 0 \\ \frac{7}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



Does this program halt?

Affine program

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

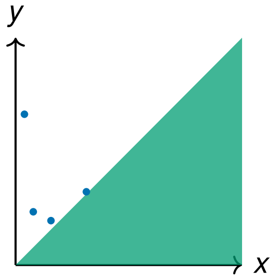


Does this program halt?

Affine program

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

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

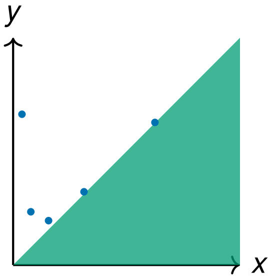


Does this program halt?

Affine program

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

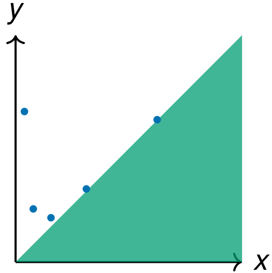
$$\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} 2 & 0 \\ \frac{7}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



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:

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

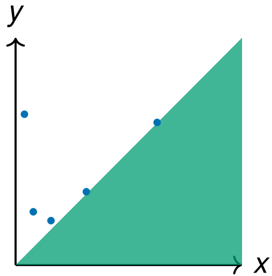
Does this program halt?

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:

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



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

Does this program halt?

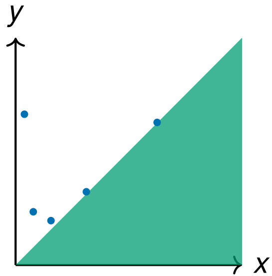
Affine program

$x := 2^{-10}$
$y := 1$
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:

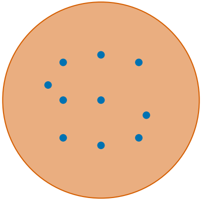
$$x^2 y - x^3 = \frac{1023}{1073741824} \tag{1}$$



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

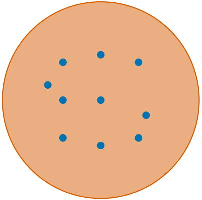
Invariants

invariant = overapproximation of the reachable states

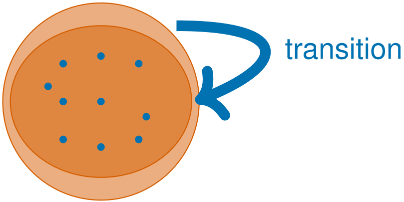


Invariants

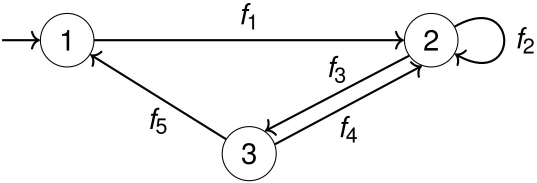
invariant = overapproximation of the reachable states



inductive invariant = invariant preserved by the transition relation

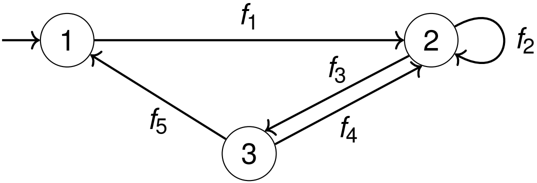


Inductive invariants: example



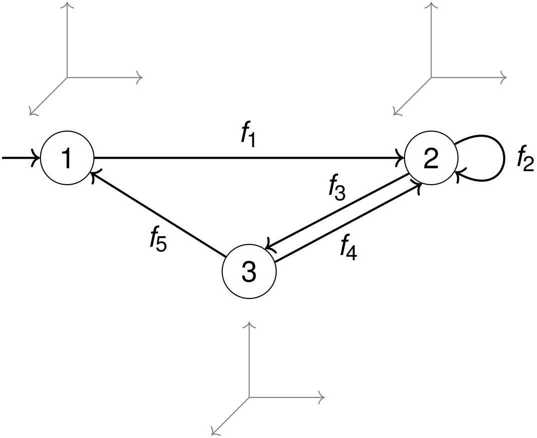
Inductive invariants: example

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



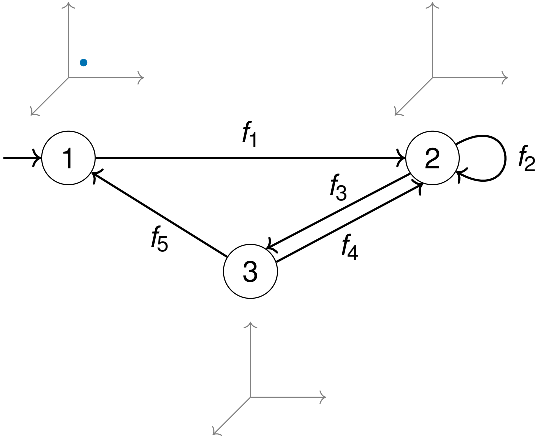
Inductive invariants: example

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



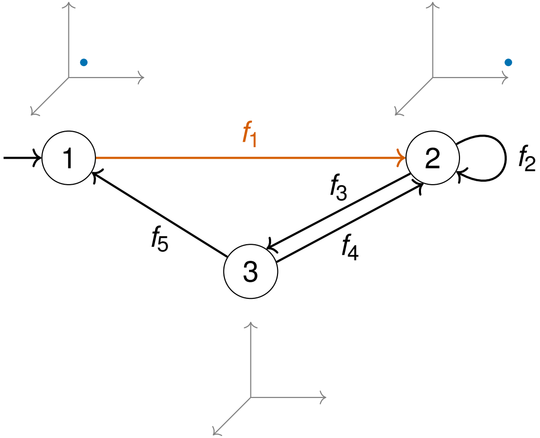
Inductive invariants: example

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



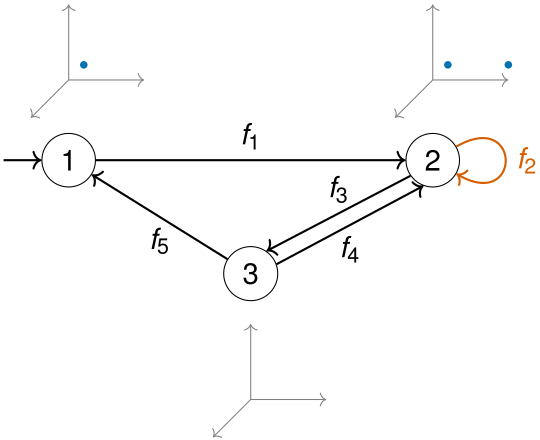
Inductive invariants: example

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



Inductive invariants: example

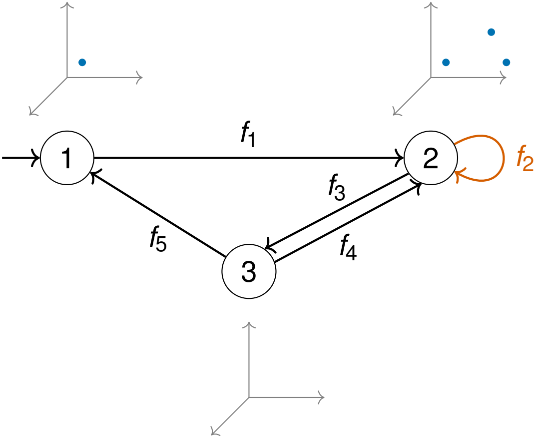
x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



Inductive invariants: example

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

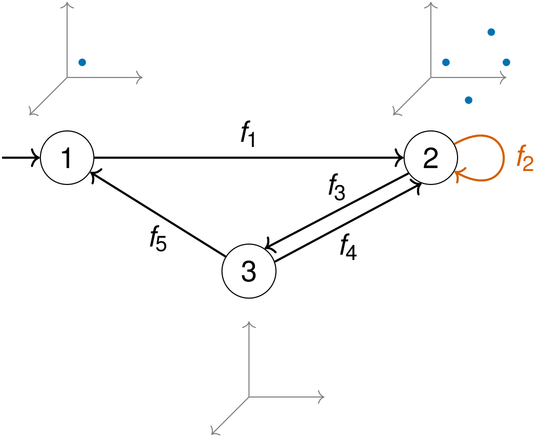
$f_i : \mathbb{R}^3 \to \mathbb{R}^3$



Inductive invariants: example

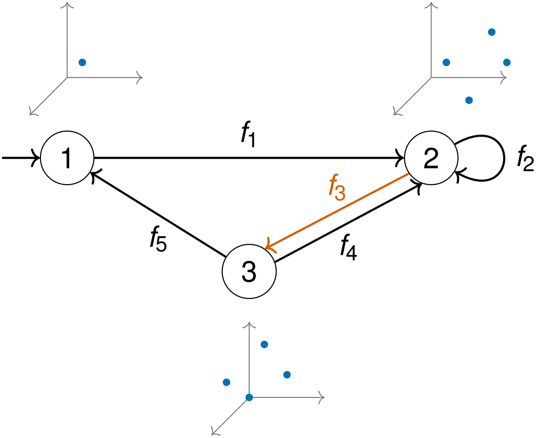
x, y, z range over $\mathbb{Q}$

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



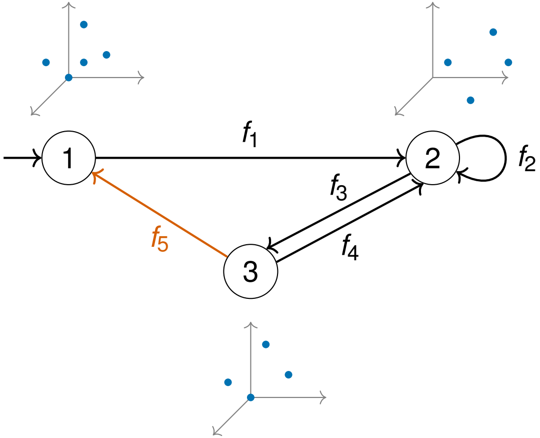
Inductive invariants: example

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



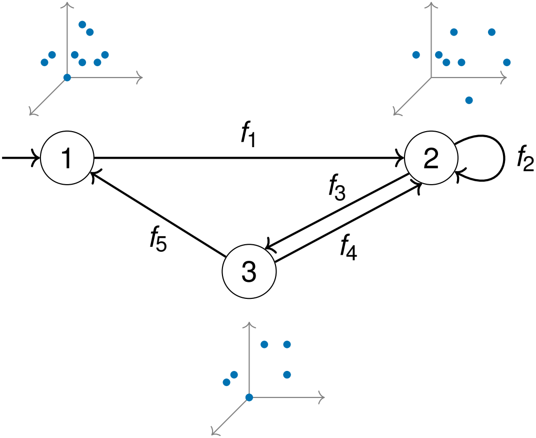
Inductive invariants: example

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



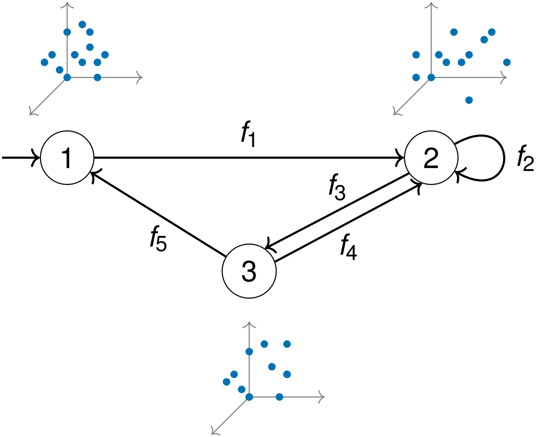
Inductive invariants: example

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



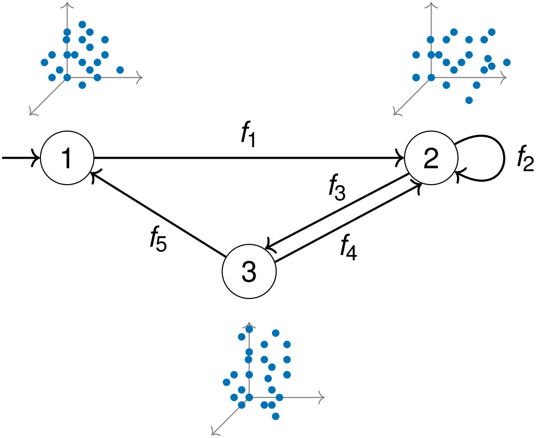
Inductive invariants: example

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



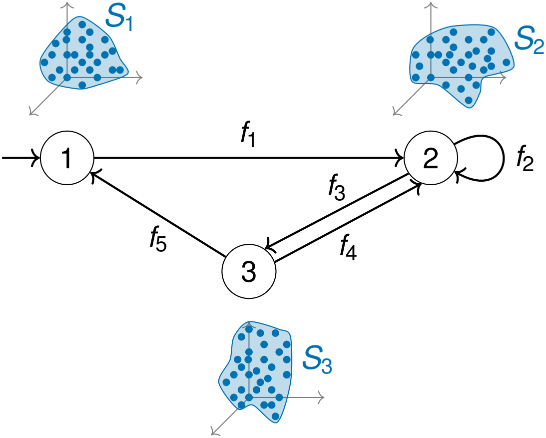
Inductive invariants: example

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



Inductive invariants: example

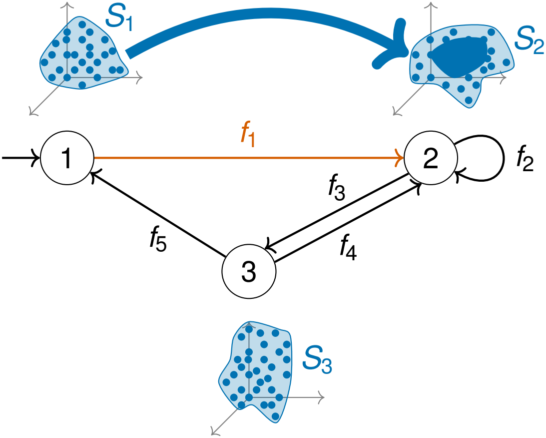
x, y, z range over $\mathbb{Q}$ $\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



S_1,S_2,S_3 are the reachable states

Inductive invariants: example

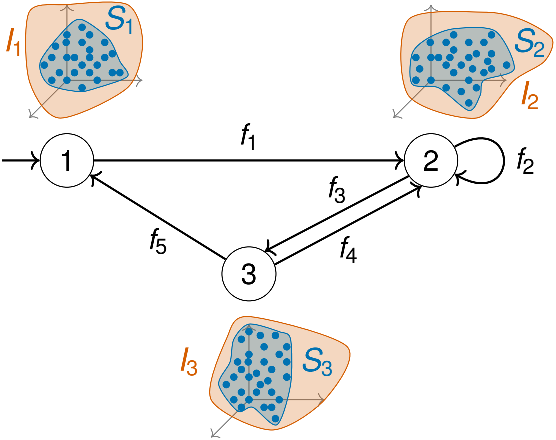
x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



S_1, S_2, S_3 is also an inductive invariant

Inductive invariants: example

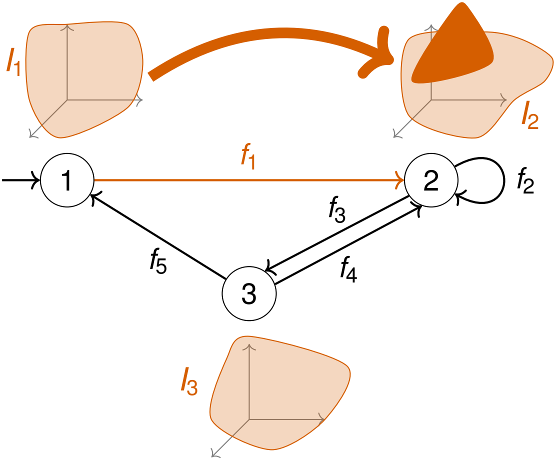
x, y, z range over $\mathbb{Q}$ $\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



I_1, I_2, I_3 is an invariant

Inductive invariants: example

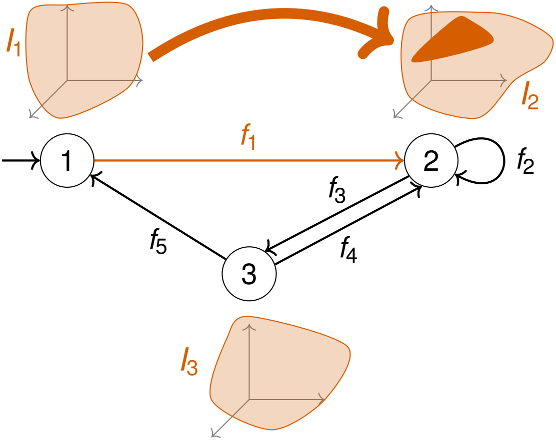
x, y, z range over $\mathbb{Q}$ $\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



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

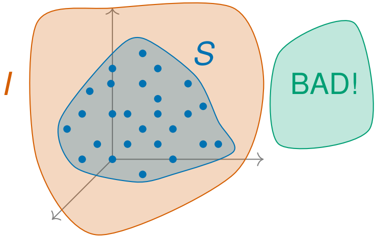
Inductive invariants: example

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

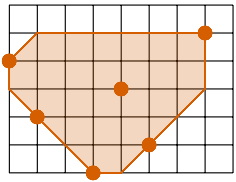
Why Invariants?



*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.***

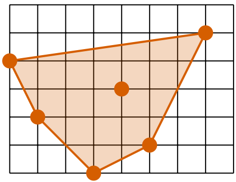
D. Beyer, T. Henzinger, R. Majumdar, and A. Rybalchenko
Invariant Synthesis for Combined Theories, 2007

Which invariants?



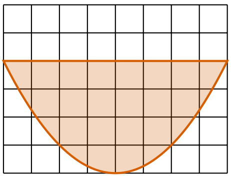
Octagons

$\leqslant$



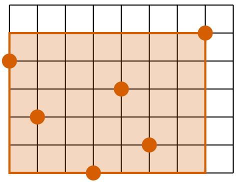
Polyhedrons

$\leqslant$

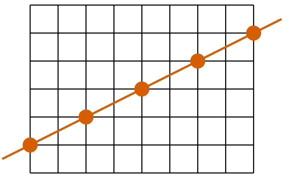


Semialgebraic sets

$\rotatebox{90}{$\leqslant$}$ $\rotatebox{90}{$\leqslant$}$ $\rotatebox{90}{$\leqslant$}$

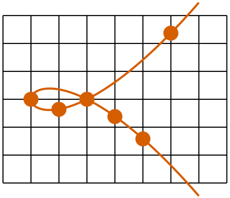


Intervals



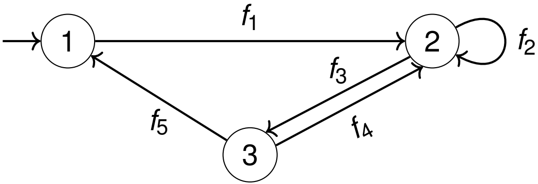
Affine/linear sets

$\leqslant$



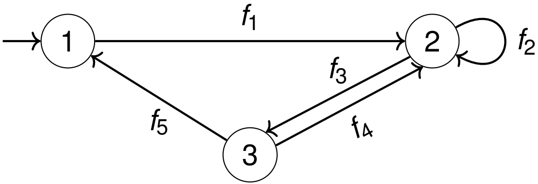
Algebraic sets = polynomial equalities

Affine programs



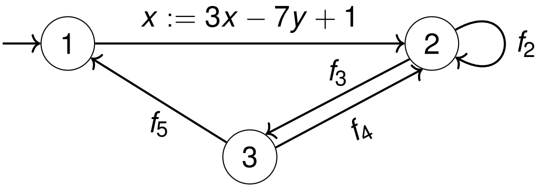
Affine programs

- Nondeterministic branching (no guards)



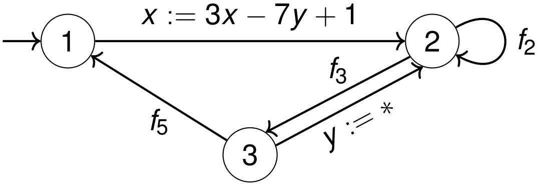
Affine programs

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



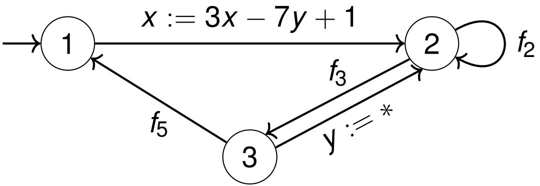
Affine programs

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



Affine programs

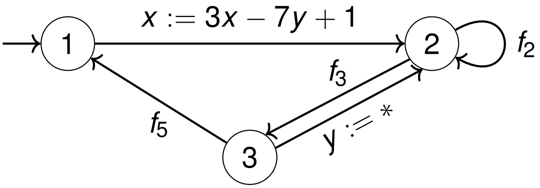
- Nondeterministic branching (no guards)
- All assignments are affine
- Allow nondeterministic assignments ($x := *$)



- 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.

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

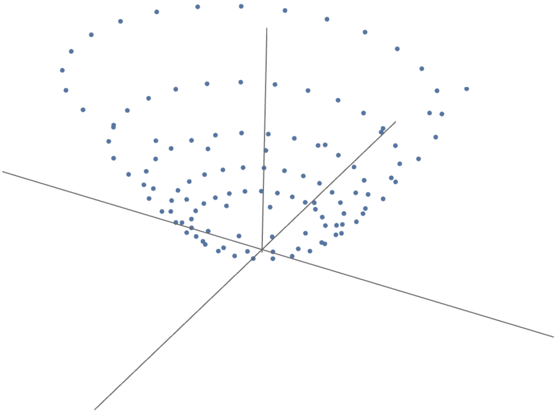
Markus Müller-Olm[1*] and Helmut Seidl[2]

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)

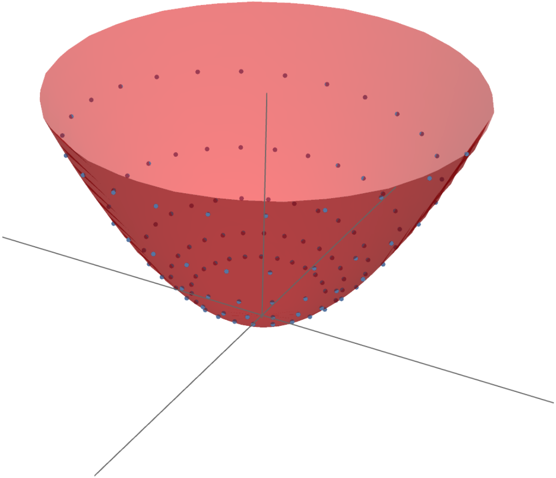
*There is an algorithm which computes, for any given affine program over $\mathbb{Q}$, all its polynomial inductive invariants up to any **fixed degree** d.*

Why fixed degree is not enough



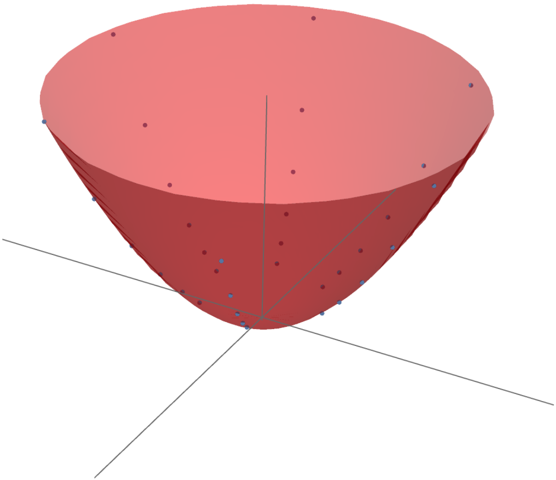
Why fixed degree is not enough

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



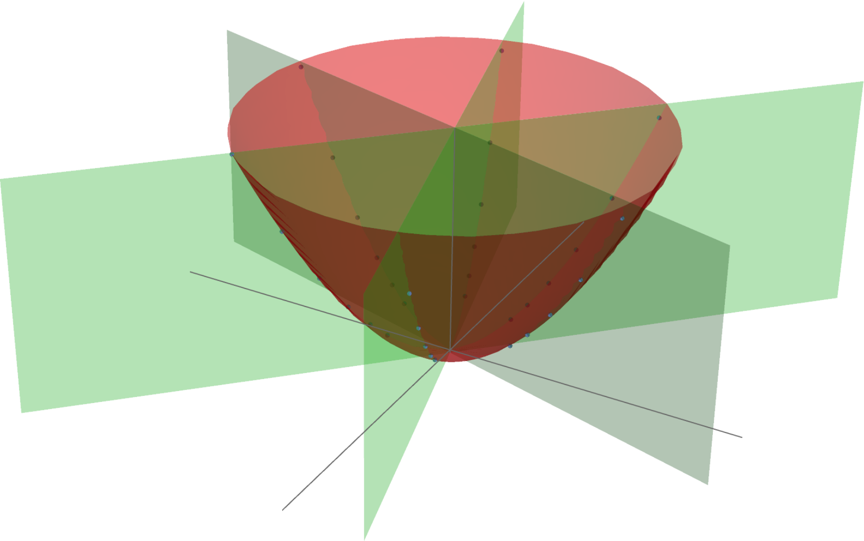
Why fixed degree is not enough

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



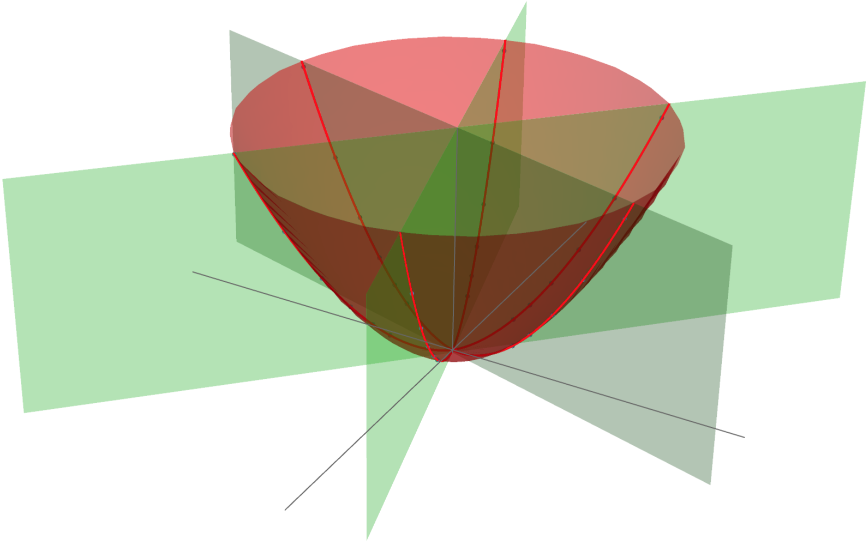
Why fixed degree is not enough

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



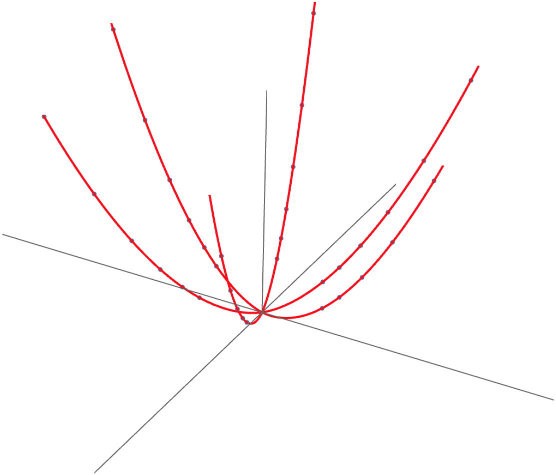
Why fixed degree is not enough

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



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 (Hrushovski, Ouaknine, P., Worrell, 2018)

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

Main result

Theorem (Hrushovski, Ouaknine, P., Worrell, 2018)

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

- strongest polynomial invariant $\iff$ smallest algebraic set

Main result

Theorem (Hrushovski, Ouaknine, P., Worrell, 2018)

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

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

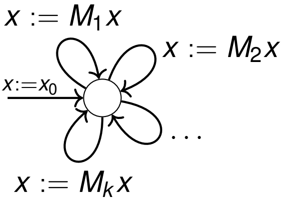
Main result

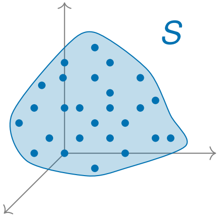
Theorem (Hrushovski, Ouaknine, P., Worrell, 2018)

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

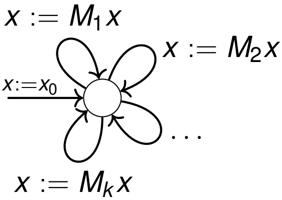
- strongest polynomial invariant $\iff$ smallest algebraic set
- Thus our algorithm computes all polynomial relations that always hold among program variables at each program location, in all possible executions of the program
- We represent this using a finite basis of polynomial equalities

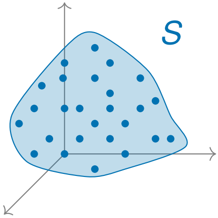
At the edge of decidability





At the edge of decidability



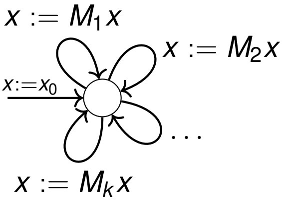


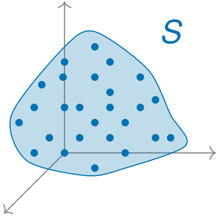
Theorem (Markov 1947*)

There is a fixed set *of* 6×6 *integer matrices* $M_1, \dots, M_k$ *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* $M_1, \dots, M_k$ *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* $M_1, \dots, M_k$*?" is undecidable for* 3×3 *matrices.*

*Original theorems about semigroups, reformulated with affine programs.

Zariski closure of finitely generated groups

Our algorithm relies on this result:

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.

Equivalently, compute the Zariski closure of a finitely generated groups of matrices.

From groups to semigroup

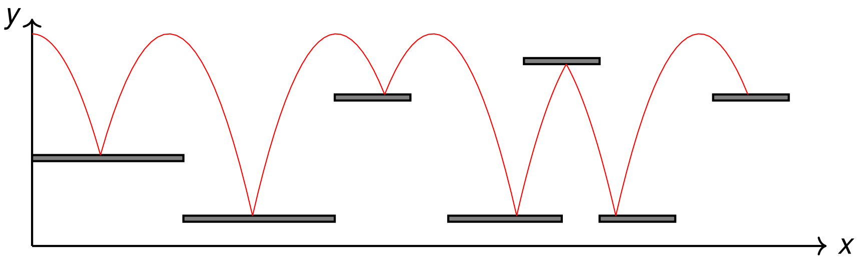
Theorem (Hrushovski, Ouaknine, P., Worrell, 2018)

There is an algorithm that computes the Zariski closure of any finitely semigroup of matrices (with algebraic coefficients), given its generators as inputs.

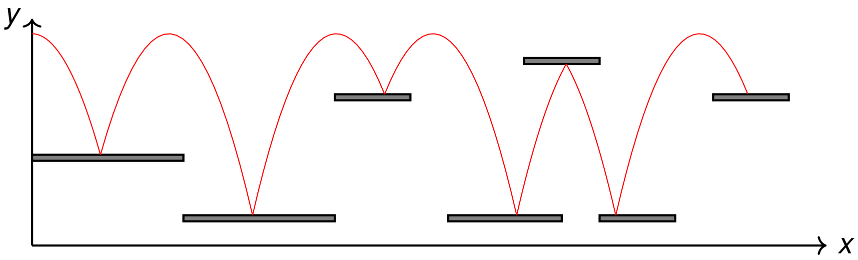
Corollary

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

Going hybrid: a bouncing ball



Going hybrid: a bouncing ball



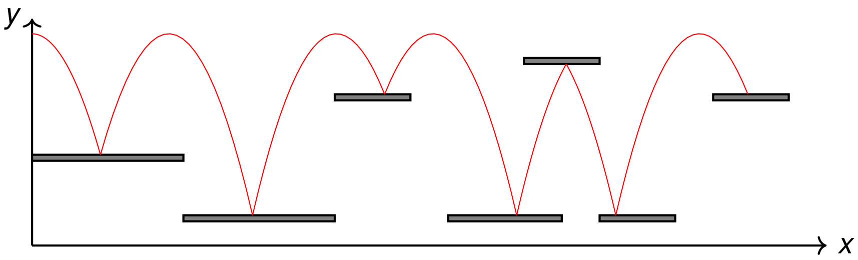
$v_y := -v_y$

$$t := 0 \quad x := 0 \quad y := h \quad v_x := c \quad v_y := 0$$

$$\boxed{\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{v}_x &= 0 \\ \dot{v}_y &= -g \\ \dot{t} &= 1 \end{aligned}}$$

- affine program: collision
- \+ linear differential equation: mechanics
- = linear hybrid automaton

Going hybrid: a bouncing ball



$v_y := -v_y$

$$t := 0 \quad x := 0 \quad y := h \quad v_x := c \quad v_y := 0$$

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{v}_x &= 0 \\ \dot{v}_y &= -g \\ \dot{t} &= 1 \end{aligned}$$

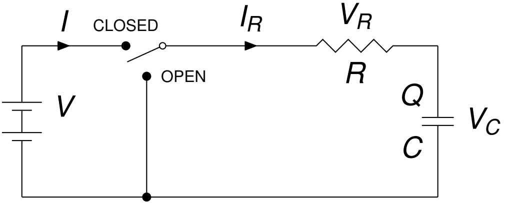
- affine program: collision
- \+ linear differential equation: mechanics
- = linear hybrid automaton

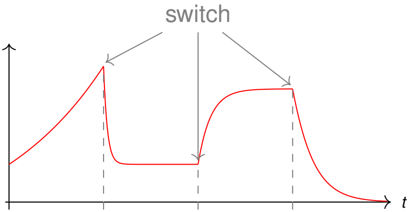
Invariants:

- $v_x = c$
- $x = tc$
- $v_y^2 + 2g(y - h) = 0$

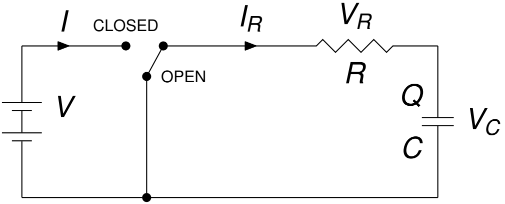
recover conservation of energy!

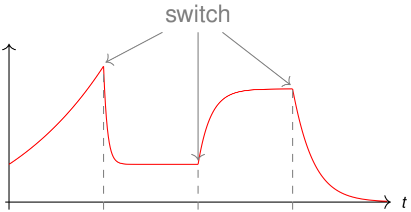
Example: RC circuit





Example: RC circuit

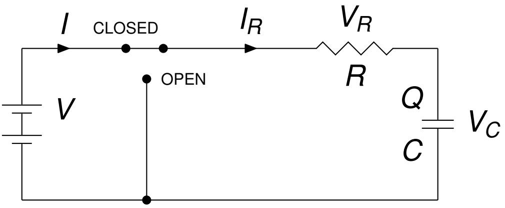


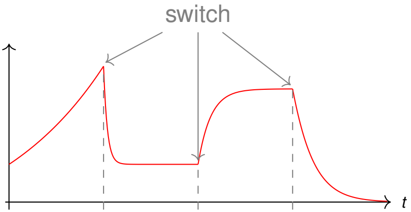


OPEN

$$\boxed{\begin{aligned} \dot{I} &= 0 \\ \dot{I}_R &= -\frac{1}{RC} I_R \\ \dot{V}_R &= -\frac{1}{C} I_R \\ \dot{Q} &= I_R \\ \dot{V}_C &= \frac{1}{C} I_R \end{aligned}}$$

Example: RC circuit





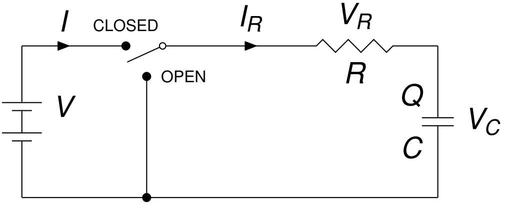
OPEN

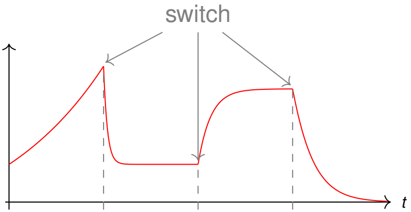
$$\boxed{\begin{aligned} \dot{I} &= 0 \\ \dot{I}_R &= -\frac{1}{RC} I_R \\ \dot{V}_R &= -\frac{1}{C} I_R \\ \dot{Q} &= I_R \\ \dot{V}_C &= \frac{1}{C} I_R \end{aligned}}$$

CLOSED

$$\boxed{\begin{aligned} \dot{I} &= -\frac{1}{RC} I_R \\ \dot{I}_R &= -\frac{1}{RC} I_R \\ \dot{V}_R &= -\frac{1}{C} I_R \\ \dot{Q} &= I_R \\ \dot{V}_C &= \frac{1}{C} I_R \end{aligned}}$$

Example: RC circuit





OPEN

$$
\begin{aligned}
\dot{I} &= 0\\
\dot{I}_R &= -\frac{1}{RC} I_R\\
\dot{V}_R &= -\frac{1}{C} I_R\\
\dot{Q} &= I_R\\
\dot{V}_C &= \frac{1}{C} I_R
\end{aligned}
$$

OPEN → CLOSED:

$$
\begin{aligned}
I &:= \frac{1}{R}(V - V_C)\\
I_R &:= \frac{1}{R}(V - V_C)\\
V_R &:= V - V_C
\end{aligned}
$$

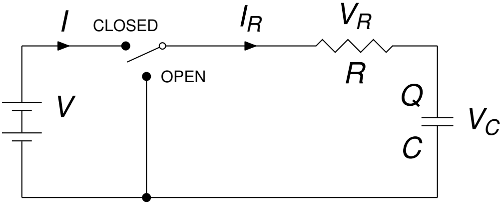
CLOSED → OPEN:

$$
\begin{aligned}
I &:= 0\\
I_R &:= -\frac{1}{R} V_C\\
V_R &:= -V_C
\end{aligned}
$$

CLOSED

$$
\begin{aligned}
\dot{I} &= -\frac{1}{RC} I_R\\
\dot{I}_R &= -\frac{1}{RC} I_R\\
\dot{V}_R &= -\frac{1}{C} I_R\\
\dot{Q} &= I_R\\
\dot{V}_C &= \frac{1}{C} I_R
\end{aligned}
$$

Example: RC circuit



Invariants

OPEN	CLOSED
$Q = CV_C$	$Q = CV_C$
$V_R = RI_R$	$V_R = RI_R$
$I = 0$	$I = I_R$
$V_R = -V_C$	$V_R = V - V_C$

OPEN

$$\begin{aligned} \dot{I} &= 0 \\ \dot{I}_R &= -\frac{1}{RC} I_R \\ \dot{V}_R &= -\frac{1}{C} I_R \\ \dot{Q} &= I_R \\ \dot{V}_C &= \frac{1}{C} I_R \end{aligned}$$

CLOSED

$$\begin{aligned} \dot{I} &= -\frac{1}{RC} I_R \\ \dot{I}_R &= -\frac{1}{RC} I_R \\ \dot{V}_R &= -\frac{1}{C} I_R \\ \dot{Q} &= I_R \\ \dot{V}_C &= \frac{1}{C} I_R \end{aligned}$$

OPEN → CLOSED:

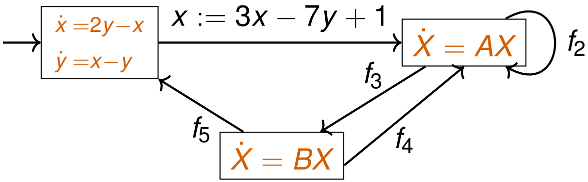
$$\begin{aligned} I &:= \frac{1}{R}(V - V_C) \\ I_R &:= \frac{1}{R}(V - V_C) \\ V_R &:= V - V_C \end{aligned}$$

CLOSED → OPEN:

$$\begin{aligned} I &:= 0 \\ I_R &:= -\frac{1}{R} V_C \\ V_R &:= -V_C \end{aligned}$$

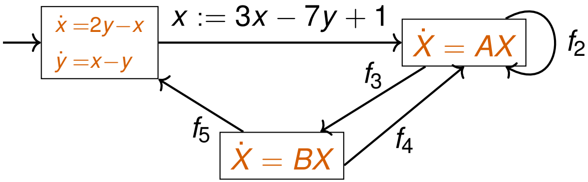
Linear Hybrid Automata

- Nondeterministic branching (no guards)
- All assignments are affine
- Linear differential equations in each location



Linear Hybrid Automata

- Nondeterministic branching (no guards)
- All assignments are affine
- Linear differential equations in each location



- More general than affine programs
- More general than linear differential equations

From affine programs to hybrid automata

Theorem (Majumdar, Ouaknine, P., Worrell, 2020)

There is an algorithm that computes, for any given guard-free linear hybrid automaton over $\overline{\mathbb{Q}}$, its strongest polynomial inductive invariant.

From affine programs to hybrid automata

Theorem (Majumdar, Ouaknine, P., Worrell, 2020)

There is an algorithm that computes, for any given guard-free linear hybrid automaton over $\overline{\mathbb{Q}}$, its strongest polynomial inductive invariant.

For systems with purely continuous dynamics, *i.e.* no discrete transitions, called switching systems:

Theorem (Hrushovski, Ouaknine, P., Worrell, 2018)

There is no algorithm that computes the strongest algebraic inductive invariant for the class of switching systems with equality guards.

From hybrid automata to affine programs

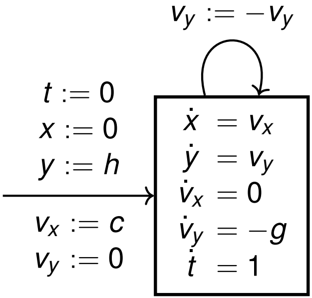
Theorem (Majumdar, Ouaknine, P., Worrell, 2020)

There is an algorithm that computes, for any given ***guard-free linear hybrid automaton*** *over $\mathbb{Q}$, an* ***affine program*** *over $\mathbb{Q}$ that has the same polynomial inductive invariants.*

From hybrid automata to affine programs

Theorem (Majumdar, Ouaknine, P., Worrell, 2020)

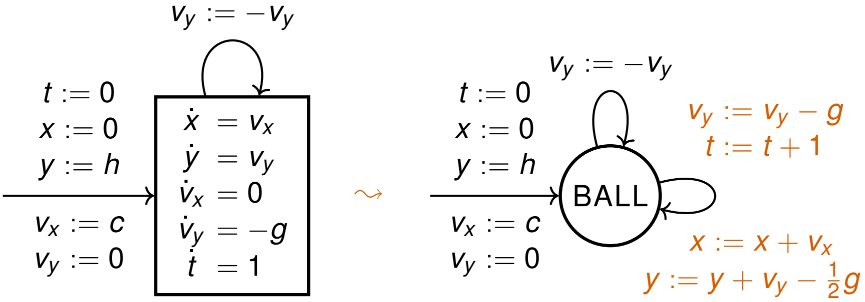
*There is an algorithm that computes, for any given **guard-free linear hybrid automaton** over $\mathbb{Q}$, an **affine program** over $\mathbb{Q}$ that has the same polynomial inductive invariants.*



From hybrid automata to affine programs

Theorem (Majumdar, Ouaknine, P., Worrell, 2020)

There is an algorithm that computes, for any given ***guard-free linear hybrid automaton*** *over $\mathbb{Q}$, an* ***affine program*** *over $\mathbb{Q}$ that has the same polynomial inductive invariants.*



Linear Differential Equations

For $x(t) \in \mathbb{R}^n$ and A rational matrix, consider

$$\dot{x} = Ax$$

The solution is

$$x(t) = e^{At}x(0)$$

where e^X is the matrix exponential.

Linear Differential Equations

For $x(t) \in \mathbb{R}^n$ and A rational matrix, consider

$$\dot{x} = Ax$$

The solution is

$$x(t) = e^{At}x(0)$$

where e^X is the matrix exponential. Recall that:

- strongest algebraic invariant = smallest algebraic set
- smallest algebraic set containing X = Zariski closure $\overline{X}$ of X

Lemma

Let A be a rational matrix, there exists B an algebraic matrix such that $\overline{\langle B \rangle} = \overline{\langle e^A \rangle} = \overline{\{e^{At} : t \in \mathbb{R}\}}$.

Linear Differential Equations

For $x(t) \in \mathbb{R}^n$ and A rational matrix, consider

$$\dot{x} = Ax$$

The solution is

$$x(t) = e^{At}x(0)$$

where e^X is the matrix exponential. Recall that:

- strongest algebraic invariant = smallest algebraic set
- smallest algebraic set containing X = Zariski closure $\overline{X}$ of X

Lemma

Let A be a rational matrix, there exists B an algebraic matrix such that $\overline{\langle B \rangle} = \overline{\langle e^A \rangle} = \overline{\{e^{At} : t \in \mathbb{R}\}}$.

- obvious candidate $B = e^A$ is not algebraic
- "reverse-engineer" B algebraic to encode some multiplicative relations between the eigenvalues

Complexity of computing the Zariski closure

How expensive is it to compute this strongest invariant ?

$$\boxed{\text{linear hybrid}} \xrightarrow{\text{reduce}} \boxed{\text{semigroup}} \xrightarrow{\text{reduce}} \boxed{\text{group}}$$

Complexity of computing the Zariski closure

How expensive is it to compute this strongest invariant ?

$$\boxed{\text{linear hybrid}} \xrightarrow{\text{reduce}} \boxed{\text{semigroup}} \xrightarrow{\text{reduce}} \boxed{\text{group}}$$

Theorem (Derksen, Jeandel and Koiran, 2004)

There is an algorithm that computes the Zariski closure of any finitely group of matrices, given its generators as inputs.

No complexity bounds. It is not clear it is even elementary.

Complexity of computing the Zariski closure

How expensive is it to compute this strongest invariant ?

$$\boxed{\text{linear hybrid}} \xrightarrow{\text{reduce}} \boxed{\text{semigroup}} \xrightarrow{\text{reduce}} \boxed{\text{group}}$$

Theorem (Derksen, Jeandel and Koiran, 2004)

There is an algorithm that computes the Zariski closure of any finitely group of matrices, given its generators as inputs.

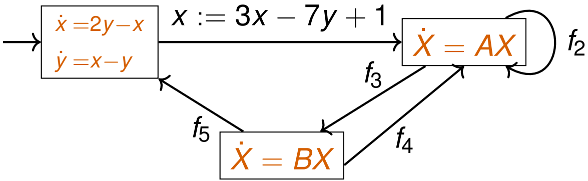
No complexity bounds. It is not clear it is even elementary.

Theorem (Nosan, P., Schmitz, Shirmohammadi, Worrell, 2022)

Given a finite set S of invertible matrices of dimension n, the algebraic group $G := \overline{\langle S \rangle}$ can be defined with equations of degree at most septuply exponential in n.

Summary

- invariant = overapproximation of reachable states
- invariants allow verification of safety properties
- guard-free linear hybrid automata:
  - nondeterministic branching, no guards, affine assignments
  - linear differential equations



Theorem (Majumdar, Ouaknine, P., Worrell, 2020)

There is an algorithm that computes, for any given guard-free linear hybrid automaton over $\mathbb{Q}$, its strongest polynomial inductive invariant.