On the complexity of some reachability problems

Amaury Pouly
Joint work with H. Bazille, O. Bournez, W. Gomaa, D. Graça, J. Ouakine, J. Worrel, J. Sousa-Pinto

10 february 2016

Outline

Piecewise Affine Systems

Polynomial Initial Value Problem

Linear hybrid automata

Piecewise Affine System (1)

General Model

- vector space: $\mathcal{H} = \mathbb{K}^d$

Piecewise Affine System (1)

General Model

- vector space: $\mathcal{H} = \mathbb{K}^d$
- partition of the space: $\mathcal{H} = \cup_{i=1}^{m} \mathcal{H}_i$
  $\mathcal{H}_i$ = convex polyhedron $= \{x \mid M_i x \leqslant v_i\}$
  $M_i \in \mathbb{Q}^{d \times d}, v_i \in \mathbb{Q}^d$

Piecewise Affine System (1)

General Model

- vector space: $\mathcal{H} = \mathbb{K}^d$
- partition of the space: $\mathcal{H} = \cup_{i=1}^{m} \mathcal{H}_i$
  $\mathcal{H}_i =$ convex polyhedron $= \{x \mid M_i x \leqslant v_i\}$
  $M_i \in \mathbb{Q}^{d \times d}, v_i \in \mathbb{Q}^d$
- piecewise affine function $f : \mathcal{H} \to \mathcal{H}$

Piecewise Affine System (1)

General Model

- vector space: $\mathcal{H} = \mathbb{K}^d$
- partition of the space: $\mathcal{H} = \cup_{i=1}^{m} \mathcal{H}_i$
  $\mathcal{H}_i$ = convex polyhedron $= \{x \mid M_i x \leqslant v_i\}$
  $M_i \in \mathbb{Q}^{d\times d}, v_i \in \mathbb{Q}^d$
- piecewise affine function $f : \mathcal{H} \to \mathcal{H}$
  $f(x) = A_i x + b_i$ for $x \in \mathcal{H}_i$ $\quad A_i \in \mathbb{Q}^{d\times d}, b_i \in \mathbb{Q}^d$

Piecewise Affine System (1)

General Model

- vector space: $\mathcal{H} = \mathbb{K}^d$
- partition of the space: $\mathcal{H} = \cup_{i=1}^{m} \mathcal{H}_i$
  $\mathcal{H}_i =$ convex polyhedron $= \{x \mid M_i x \leqslant v_i\}$
  $M_i \in \mathbb{Q}^{d\times d}, v_i \in \mathbb{Q}^d$
- piecewise affine function $f : \mathcal{H} \to \mathcal{H}$
  $f(x) = A_i x + b_i$ for $x \in \mathcal{H}_i$ $\qquad A_i \in \mathbb{Q}^{d\times d}, b_i \in \mathbb{Q}^d$
- trajectory: $x, f(x), f^{[2]}(x), \ldots, f^{[i]}(x), \ldots$

Piecewise Affine System (1)

General Model

- vector space: $\mathcal{H} = \mathbb{K}^d$
- partition of the space: $\mathcal{H} = \cup_{i=1}^{m} \mathcal{H}_i$
  $\mathcal{H}_i =$ convex polyhedron $= \{x \mid M_i x \leqslant v_i\}$
  $M_i \in \mathbb{Q}^{d \times d}, v_i \in \mathbb{Q}^d$
- piecewise affine function $f : \mathcal{H} \to \mathcal{H}$
  $f(x) = A_i x + b_i$ for $x \in \mathcal{H}_i$ $\qquad A_i \in \mathbb{Q}^{d \times d}, b_i \in \mathbb{Q}^d$
- trajectory: $x, f(x), f^{[2]}(x), \ldots, f^{[i]}(x), \ldots$

$\Rightarrow$ Discrete time dynamical system

Piecewise Affine System (1)

General Model

- vector space: $\mathcal{H} = \mathbb{K}^d$
- partition of the space: $\mathcal{H} = \cup_{i=1}^{m} \mathcal{H}_i$
  $\mathcal{H}_i$ = convex polyhedron $= \{x \mid M_i x \leqslant v_i\}$
  $M_i \in \mathbb{Q}^{d \times d}, v_i \in \mathbb{Q}^d$
- piecewise affine function $f : \mathcal{H} \to \mathcal{H}$
  $f(x) = A_i x + b_i$ for $x \in \mathcal{H}_i$ $\quad A_i \in \mathbb{Q}^{d \times d}, b_i \in \mathbb{Q}^d$
- trajectory: $x, f(x), f^{[2]}(x), \ldots, f^{[i]}(x), \ldots$

$\Rightarrow$ Discrete time dynamical system

Three cases:

- $\mathbb{K} = \mathbb{N}$: integer case
- $\mathbb{K} = [0, 1]$: continuous bounded case
- $\mathbb{K} = \mathbb{R}$: continuous unbounded case

Piecewise Affine System (1)

General Model

- vector space: $\mathcal{H} = \mathbb{K}^d$
- partition of the space: $\mathcal{H} = \cup_{i=1}^{m} \mathcal{H}_i$
  $\mathcal{H}_i$ = convex polyhedron $= \{x \mid M_i x \leqslant v_i\}$
  $M_i \in \mathbb{Q}^{d\times d}, v_i \in \mathbb{Q}^d$
- piecewise affine function $f : \mathcal{H} \to \mathcal{H}$
  $f(x) = A_i x + b_i$ for $x \in \mathcal{H}_i$ $\quad A_i \in \mathbb{Q}^{d\times d}, b_i \in \mathbb{Q}^d$
- trajectory: $x, f(x), f^{[2]}(x), \dots, f^{[i]}(x), \dots$

$\Rightarrow$ Discrete time dynamical system

Three cases:

- $\mathbb{K} = \mathbb{N}$: integer case $\to$ Very different from $[0, 1]$ and $\mathbb{R}$
- $\mathbb{K} = [0, 1]$: continuous bounded case
- $\mathbb{K} = \mathbb{R}$: continuous unbounded case

Piecewise Affine System (1)

General Model

- vector space: $\mathcal{H} = \mathbb{K}^d$
- partition of the space: $\mathcal{H} = \cup_{i=1}^{m} \mathcal{H}_i$
  $\mathcal{H}_i =$ convex polyhedron $= \{x \mid M_i x \leqslant v_i\}$
  $M_i \in \mathbb{Q}^{d\times d}, v_i \in \mathbb{Q}^d$
- piecewise affine function $f : \mathcal{H} \to \mathcal{H}$
  $f(x) = A_i x + b_i$ for $x \in \mathcal{H}_i$ $\qquad A_i \in \mathbb{Q}^{d\times d}, b_i \in \mathbb{Q}^d$
- trajectory: $x, f(x), f^{[2]}(x), \dots, f^{[i]}(x), \dots$

$\Rightarrow$ Discrete time dynamical system

Three cases:

- $\mathbb{K} = \mathbb{N}$: integer case $\rightarrow$ Very different from $[0,1]$ and $\mathbb{R}$
- $\mathbb{K} = [0,1]$: continuous bounded case $\rightarrow$ Our case
- $\mathbb{K} = \mathbb{R}$: continuous unbounded case

Piecewise Affine System (1)

General Model

- vector space: $\mathcal{H} = \mathbb{K}^d$
- partition of the space: $\mathcal{H} = \cup_{i=1}^{m} \mathcal{H}_i$
  $\mathcal{H}_i$ = convex polyhedron $= \{x \mid M_i x \leqslant v_i\}$
  $M_i \in \mathbb{Q}^{d \times d}, v_i \in \mathbb{Q}^d$
- piecewise affine function $f : \mathcal{H} \to \mathcal{H}$
  $f(x) = A_i x + b_i$ for $x \in \mathcal{H}_i$ $\quad A_i \in \mathbb{Q}^{d \times d}, b_i \in \mathbb{Q}^d$
- trajectory: $x, f(x), f^{[2]}(x), \ldots, f^{[i]}(x), \ldots$

$\Rightarrow$ Discrete time dynamical system

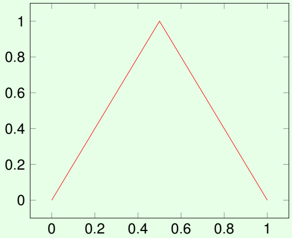
Three cases:

- $\mathbb{K} = \mathbb{N}$: integer case $\to$ Very different from $[0, 1]$ and $\mathbb{R}$
- $\mathbb{K} = [0, 1]$: continuous bounded case $\to$ Our case
- $\mathbb{K} = \mathbb{R}$: continuous unbounded case $\to$ Similar*ish* to $[0, 1]$?

Piecewise Affine System (2)

f continuous

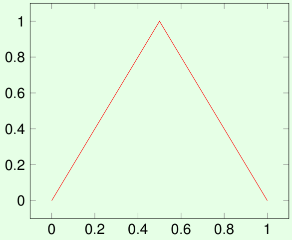
$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



Piecewise Affine System (2)

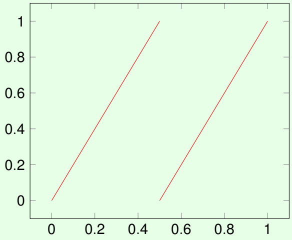
f continuous

$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



f discontinuous

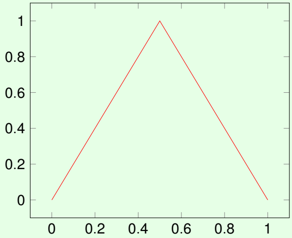
$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}[\\ 2x - 1 & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



Piecewise Affine System (2)

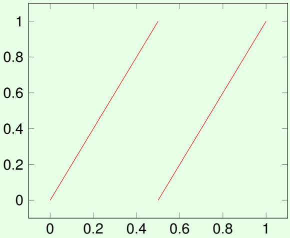
f continuous

$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



f discontinuous

$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}[\\ 2x - 1 & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$

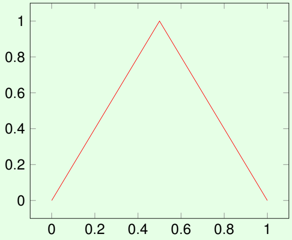


$\rightarrow$ Our case

Piecewise Affine System (2)

f continuous

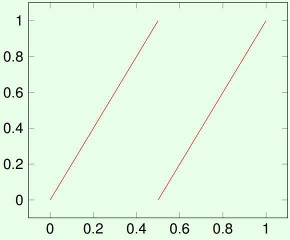
$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



$\rightarrow$ Our case

f discontinuous

$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}[\\ 2x - 1 & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$

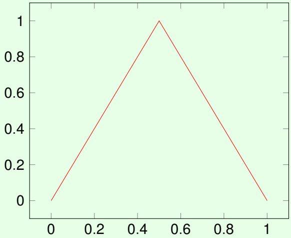


$\rightarrow$ Quite different

Example

Function

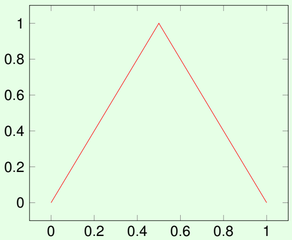
$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



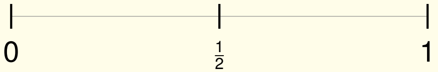
Example

Function

$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



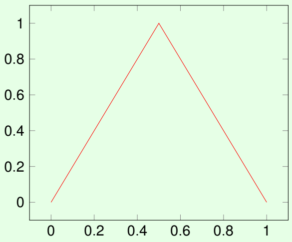
Trajectory



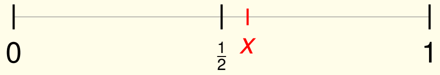
Example

Function

$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



Trajectory

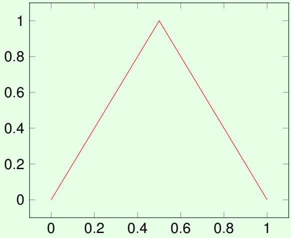


$$x = 0.5625$$

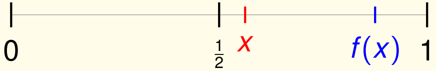
Example

Function

$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



Trajectory



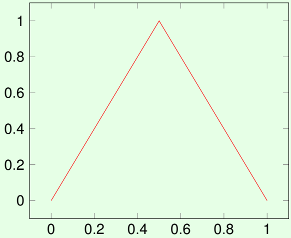
$$x = 0.5625$$

$$f(x) = 0.875$$

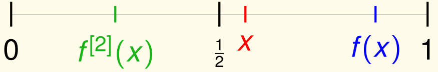
Example

Function

$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



Trajectory

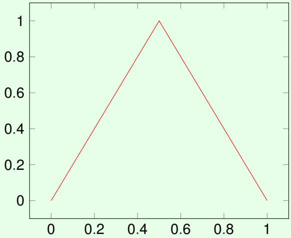


$$x = 0.5625$$
$$f(x) = 0.875$$
$$f^{[2]}(x) = 0.25$$

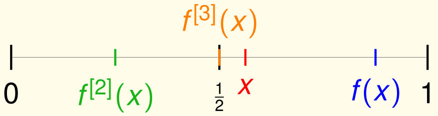
Example

Function

$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



Trajectory

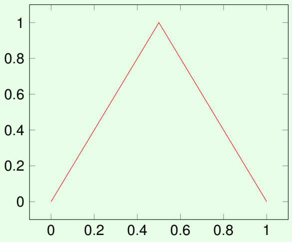


$$x = 0.5625$$
$$f(x) = 0.875$$
$$f^{[2]}(x) = 0.25$$
$$f^{[3]}(x) = 0.5$$

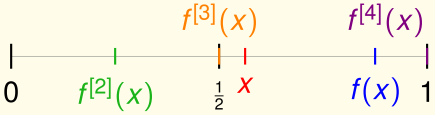
Example

Function

$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



Trajectory

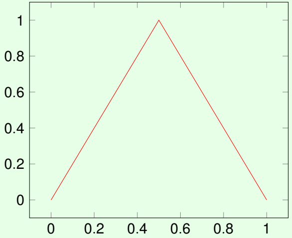


$$x = 0.5625$$
$$f(x) = 0.875$$
$$f^{[2]}(x) = 0.25$$
$$f^{[3]}(x) = 0.5$$
$$f^{[4]}(x) = 1$$

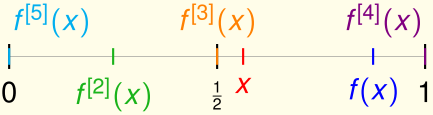
Example

Function

$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



Trajectory

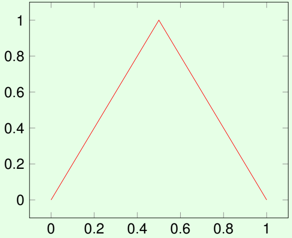


$$x = 0.5625$$
$$f(x) = 0.875$$
$$f^{[2]}(x) = 0.25$$
$$f^{[3]}(x) = 0.5$$
$$f^{[4]}(x) = 1$$
$$f^{[5]}(x) = 0$$

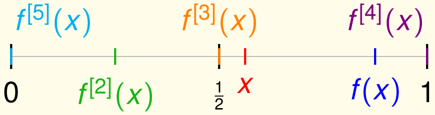
Example

Function

$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



Trajectory

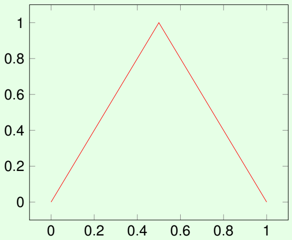


$$\begin{aligned} x &= 0.5625 \\ f(x) &= 0.875 \\ f^{[2]}(x) &= 0.25 \\ f^{[3]}(x) &= 0.5 \\ f^{[4]}(x) &= 1 \\ f^{[n]}(x) &= 0 \qquad n \geqslant 5 \end{aligned}$$

Example

Function

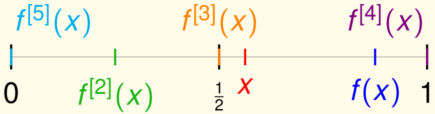
$$f(x) = \begin{cases} 2x & \text{if } x \in [0, \frac{1}{2}] \\ 2 - 2x & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



Remark

Trajectory depends on the binary expansion of x

Trajectory



$$x = 0.5625$$

$$f(x) = 0.875$$

$$f^{[2]}(x) = 0.25$$

$$f^{[3]}(x) = 0.5$$

$$f^{[4]}(x) = 1$$

$$f^{[n]}(x) = 0 \qquad n \geqslant 5$$

Existings Results

Problem: `REACH-REGION`

- **Input:** $f : [0,1]^d \to [0,1]^d$ continuous, piecewise affine

Existings Results

Problem: `REACH-REGION`

- **Input:** $f : [0,1]^d \rightarrow [0,1]^d$ continuous, piecewise affine
- **Input:** R_0, R: convex regions of $[0,1]^d$

Existings Results

Problem: `REACH-REGION`

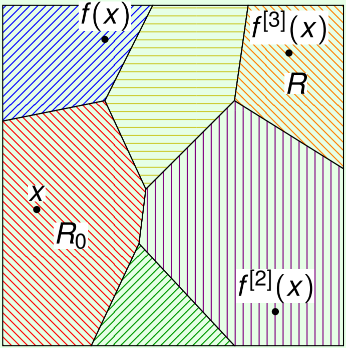
- **Input:** $f : [0,1]^d \to [0,1]^d$ continuous, piecewise affine
- **Input:** R_0, R: convex regions of $[0,1]^d$
- **Question:** $\exists x \in R_0, \exists t \in \mathbb{N}, f^{[t]}(x) \in R$?

Existings Results

Problem: REACH-REGION

- **Input:** $f : [0,1]^d \to [0,1]^d$ continuous, piecewise affine
- **Input:** R_0, R: convex regions of $[0,1]^d$
- **Question:** $\exists x \in R_0, \exists t \in \mathbb{N}, f^{[t]}(x) \in R$?

Example

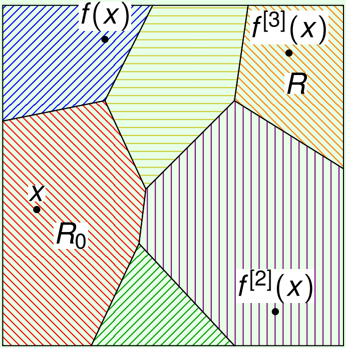


Existings Results

Problem: REACH−REGION

- **Input:** $f : [0,1]^d \to [0,1]^d$ continuous, piecewise affine
- **Input:** R_0, R: convex regions of $[0,1]^d$
- **Question:** $\exists x \in R_0, \exists t \in \mathbb{N}, f^{[t]}(x) \in R$?

Example



Theorem (Koiran, Cosnard, Garzon)

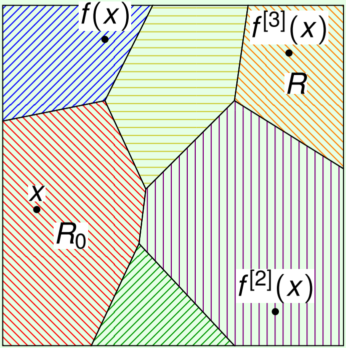
REACH−REGION is undecidable for $d \geqslant 2$

Existings Results

Problem: REACH−REGION

- **Input:** $f : [0,1]^d \to [0,1]^d$ continuous, piecewise affine
- **Input:** R_0, R: convex regions of $[0,1]^d$
- **Question:** $\exists x \in R_0, \exists t \in \mathbb{N}, f^{[t]}(x) \in R$?

Example



Theorem (Koiran, Cosnard, Garzon)

REACH−REGION is undecidable for $d \geqslant 2$

Proof (Idea)

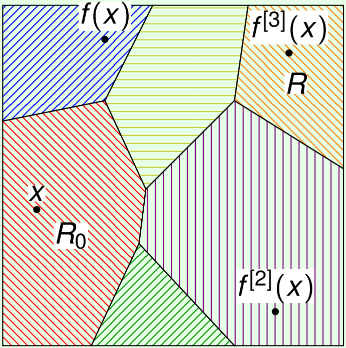
Simulate a Turing Machine and reduce from halting problem.

Existings Results

Problem: REACH−REGION

- **Input:** $f : [0,1]^d \to [0,1]^d$ continuous, piecewise affine
- **Input:** R_0, R: convex regions of $[0,1]^d$
- **Question:** $\exists x \in R_0, \exists t \in \mathbb{N}, f^{[t]}(x) \in R$?

Example



Theorem (Koiran, Cosnard, Garzon)

REACH−REGION is undecidable for $d \geqslant 2$

Proof (Idea)

Simulate a Turing Machine and reduce from halting problem.

Open Problem

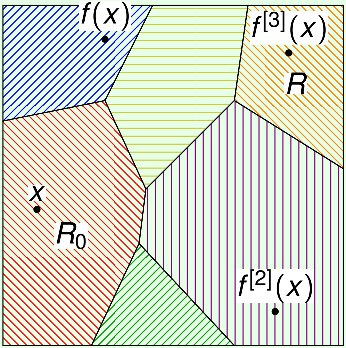
Decidability for $d = 1$, even for two intervals.

Existings Results

Problem: CONTROL−REGION

- **Input:** $f : [0,1]^d \to [0,1]^d$ continuous, piecewise affine
- **Input:** R_0, R: convex regions of $[0,1]^d$
- **Question:** $\forall x \in R_0, \exists t \in \mathbb{N}, f^{[t]}(x) \in R$?

Example



Theorem (Blondel, Bournez, Koiran, Tsitsiklis)

CONTROL−REGION is undecidable for $d \geqslant 2$

Proof (Idea)

Harder simulation of a Turing Machine

Open Problem

Decidability for $d = 1$, even for two intervals.

Our Results

Problem: REACH-REGION-TIME

- **Input:** $f : [0,1]^d \rightarrow [0,1]^d$ continuous, piecewise affine

Our Results

Problem: REACH-REGION-TIME

- **Input:** $f : [0,1]^d \to [0,1]^d$ continuous, piecewise affine
- **Input:** R_0, R: convex regions of $[0,1]^d$, $T \in \mathbb{N}$ in unary

Our Results

Problem: REACH-REGION-TIME

- **Input:** $f : [0,1]^d \to [0,1]^d$ continuous, piecewise affine
- **Input:** R_0, R: convex regions of $[0,1]^d$, $T \in \mathbb{N}$ in unary
- **Question:** $\exists x \in R_0, \exists t \leqslant T, f^{[t]}(x) \in R$?

Our Results

Problem: REACH-REGION-TIME

- **Input:** $f : [0,1]^d \to [0,1]^d$ continuous, piecewise affine
- **Input:** R_0, R: convex regions of $[0,1]^d$, $T \in \mathbb{N}$ in unary
- **Question:** $\exists x \in R_0, \exists t \leqslant T, f^{[t]}(x) \in R$?

Theorem

REACH-REGION-TIME is NP-complete for $d \geqslant 2$

Our Results

Problem: REACH-REGION-TIME

- **Input:** $f : [0,1]^d \to [0,1]^d$ continuous, piecewise affine
- **Input:** R_0, R: convex regions of $[0,1]^d$, $T \in \mathbb{N}$ in unary
- **Question:** $\exists x \in R_0, \exists t \leqslant T, f^{[t]}(x) \in R$?

Theorem

REACH-REGION-TIME is NP-complete for $d \geqslant 2$

Open Problem

Complexity for $d = 1$.

Our Results

Problem: CONTROL-REGION-TIME

- **Input:** $f : [0,1]^d \to [0,1]^d$ continuous, piecewise affine
- **Input:** R_0, R: convex regions of $[0,1]^d$, $T \in \mathbb{N}$ in unary
- **Question:** $\forall x \in R_0, \exists t \leqslant T, f^{[t]}(x) \in R$?

Theorem

CONTROL-REGION-TIME is coNP-complete for $d \geqslant 2$

Open Problem

Complexity for $d = 1$.

REACH−REGION−TIME is in NP

Proof.

Given f, R_0, $R = R_n$ and T:

REACH-REGION-TIME is in NP

Proof.

Given f, R_0, $R = R_n$ and T:

- Guess reach time $t \leqslant T$ $\leftarrow$ Nondeterministic poly (NP)

REACH-REGION-TIME is in NP

Proof.

Given f, R_0, $R = R_n$ and T:

- Guess reach time $t \leqslant T$ ← Nondeterministic poly (NP)
- Guess sequence of regions $r_1, \ldots, r_{t-1}$ ← NP

REACH-REGION-TIME is in NP

Proof.

Given f, R_0, $R = R_n$ and T:

- Guess reach time $t \leqslant T$ ← Nondeterministic poly (NP)
- Guess sequence of regions $r_1, \ldots, r_{t-1}$ ← NP
- Guess starting point $x \in \mathbb{Q}^d$ of polynomial size ← NP

REACH−REGION−TIME is in NP

Proof.

Given f, R_0, $R = R_n$ and T:

- Guess reach time $t \leqslant T$ ← Nondeterministic poly (NP)
- Guess sequence of regions $r_1, \dots, r_{t-1}$ ← NP
- Guess starting point $x \in \mathbb{Q}^d$ of polynomial size ← NP
- Check that $f^{[i]}(x) \in R_{r_i}$ for all $i \in \{0, \dots, t\}$:

REACH-REGION-TIME is in NP

Proof.

Given f, R_0, $R = R_n$ and T:

- Guess reach time $t \leqslant T$ ← Nondeterministic poly (NP)
- Guess sequence of regions $r_1, \ldots, r_{t-1}$ ← NP
- Guess starting point $x \in \mathbb{Q}^d$ of polynomial size ← NP
- Check that $f^{[i]}(x) \in R_{r_i}$ for all $i \in \{0, \ldots, t\}$:
  $f^{[i]}(x) \in R_{r_i} \Leftrightarrow M_{r_i}(C_i x + d_i) \leqslant v_i$ ← Polynomial size

REACH-REGION-TIME is in NP

Proof.

Given f, R_0, $R = R_n$ and T:

- Guess reach time $t \leqslant T$ ← Nondeterministic poly (NP)
- Guess sequence of regions $r_1, \ldots, r_{t-1}$ ← NP
- Guess starting point $x \in \mathbb{Q}^d$ of polynomial size ← NP
- Check that $f^{[i]}(x) \in R_{r_i}$ for all $i \in \{0, \ldots, t\}$:
  $f^{[i]}(x) \in R_{r_i} \Leftrightarrow M_{r_i}(C_i x + d_i) \leqslant v_i$ ← Polynomial size
- Accept if all systems are satisfied

□

REACH-REGION-TIME is in NP

Proof.

Given f, R_0, $R = R_n$ and T:

- Guess reach time $t \leqslant T$ ← Nondeterministic poly (NP)
- Guess sequence of regions $r_1, \dots, r_{t-1}$ ← NP
- Guess starting point $x \in \mathbb{Q}^d$ of polynomial size ← NP
- Check that $f^{[i]}(x) \in R_{r_i}$ for all $i \in \{0, \dots, t\}$:
  $f^{[i]}(x) \in R_{r_i} \Leftrightarrow M_{r_i}(C_i x + d_i) \leqslant v_i$ ← Polynomial size
- Accept if all systems are satisfied

□

Works because:

Every satisfiable rational linear system $Ax \leqslant b$ has a rational solution of polynomial size.

REACH-REGION-TIME is NP-hard

Reduce from:

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \ldots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \ldots, n\}, \sum_{i \in I} A_i = B$?

REACH-REGION-TIME is NP-hard

Reduce from:

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \dots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \dots, n\}, \sum_{i \in I} A_i = B$?

- Carefully chosen: simplest we could find, already “affine”

REACH-REGION-TIME is NP-hard

Reduce from:

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \dots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \dots, n\}, \sum_{i \in I} A_i = B$?

- Carefully chosen: simplest we could find, already "affine"
- Still reduction is very tricky

REACH-REGION-TIME is NP-hard (proof idea)

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \dots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \dots, n\}, \sum_{i \in I} A_i = B$?

REACH-REGION-TIME is NP-hard (proof idea)

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \dots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \dots, n\}, \sum_{i \in I} A_i = B$?

Each point (x, y) encodes a triple (σ, i, I):

- σ: current sum, i: current index
- I: subset of $\{1, \dots, n\}$

REACH-REGION-TIME is NP-hard (proof idea)

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \dots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \dots, n\}, \sum_{i \in I} A_i = B$?

Each point (x, y) encodes a triple (σ, i, I):

- σ: current sum, i: current index
- I: subset of $\{1, \dots, n\}$

One step of f on (encoded) (σ, i, I) does:

- $\sigma \mapsto \sigma + A_i$ if $i \in I$, otherwise unchanged
- $i \mapsto i + 1$ and $I \mapsto I \setminus \{i\}$

REACH-REGION-TIME is NP-hard (proof idea)

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \ldots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \ldots, n\}, \sum_{i \in I} A_i = B$?

Each point (x, y) encodes a triple (σ, i, I):

- σ: current sum, i: current index
- I: subset of $\{1, \ldots, n\}$

One step of f on (encoded) (σ, i, I) does:

- $\sigma \mapsto \sigma + A_i$ if $i \in I$, otherwise unchanged
- $i \mapsto i + 1$ and $I \mapsto I \setminus \{i\}$

Initial points:

- $\sigma = 0$ and $i = 1$
- I: any possible subset of $\{1, \ldots, n\}$

REACH-REGION-TIME is NP-hard (proof idea)

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \dots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \dots, n\}, \sum_{i \in I} A_i = B$?

Each point (x, y) encodes a triple (σ, i, I):

- σ: current sum, i: current index
- I: subset of $\{1, \dots, n\}$

One step of f on (encoded) (σ, i, I) does:

- $\sigma \mapsto \sigma + A_i$ if $i \in I$, otherwise unchanged
- $i \mapsto i + 1$ and $I \mapsto I \setminus \{i\}$

Initial points:

- $\sigma = 0$ and $i = 1$
- I: any possible subset of $\{1, \dots, n\}$

Example with $A_1 = 42, A_2 = 13$ and $A_3 = 7$:

$(0, 1, \{1, 3\}) \mapsto$

REACH-REGION-TIME is NP-hard (proof idea)

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \ldots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \ldots, n\}, \sum_{i \in I} A_i = B$?

Each point (x, y) encodes a triple (σ, i, I):

- σ: current sum, i: current index
- I: subset of $\{1, \ldots, n\}$

One step of f on (encoded) (σ, i, I) does:

- $\sigma \mapsto \sigma + A_i$ if $i \in I$, otherwise unchanged
- $i \mapsto i + 1$ and $I \mapsto I \setminus \{i\}$

Initial points:

- $\sigma = 0$ and $i = 1$
- I: any possible subset of $\{1, \ldots, n\}$

Example with $A_1 = 42, A_2 = 13$ and $A_3 = 7$:

$(0, 1, \{1, 3\}) \mapsto (42, 2, \{3\}) \mapsto$

REACH-REGION-TIME is NP-hard (proof idea)

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \dots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \dots, n\}, \sum_{i \in I} A_i = B$?

Each point (x, y) encodes a triple (σ, i, I):

- σ: current sum, i: current index
- I: subset of $\{1, \dots, n\}$

One step of f on (encoded) (σ, i, I) does:

- $\sigma \mapsto \sigma + A_i$ if $i \in I$, otherwise unchanged
- $i \mapsto i + 1$ and $I \mapsto I \setminus \{i\}$

Initial points:

- $\sigma = 0$ and $i = 1$
- I: any possible subset of $\{1, \dots, n\}$

Example with $A_1 = 42, A_2 = 13$ and $A_3 = 7$:

$$(0, 1, \{1, 3\}) \mapsto (42, 2, \{3\}) \mapsto (42, 3, \{3\}) \mapsto$$

REACH-REGION-TIME is NP-hard (proof idea)

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \dots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \dots, n\}, \sum_{i \in I} A_i = B$?

Each point (x, y) encodes a triple (σ, i, I):

- σ: current sum, i: current index
- I: subset of $\{1, \dots, n\}$

One step of f on (encoded) (σ, i, I) does:

- $\sigma \mapsto \sigma + A_i$ if $i \in I$, otherwise unchanged
- $i \mapsto i + 1$ and $I \mapsto I \setminus \{i\}$

Initial points:

- $\sigma = 0$ and $i = 1$
- I: any possible subset of $\{1, \dots, n\}$

Example with $A_1 = 42, A_2 = 13$ and $A_3 = 7$:

$$(0, 1, \{1, 3\}) \mapsto (42, 2, \{3\}) \mapsto (42, 3, \{3\}) \mapsto (49, 4, \varnothing)$$

$\texttt{REACH-REGION-TIME}$ is NP-hard (proof idea)

Problem $\texttt{SUBSET-SUM}$

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \ldots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \ldots, n\}, \sum_{i \in I} A_i = B$?

Each point (x, y) encodes a triple (σ, i, I):

$$x = i2^{-k} + \sigma 2^{-\ell} \qquad y = \sum_{i \in I} 2^{-i}$$

REACH-REGION-TIME is NP-hard (proof idea)

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \dots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \dots, n\}, \sum_{i \in I} A_i = B$?

Each point (x, y) encodes a triple (σ, i, I):

$$x = i2^{-k} + \sigma 2^{-\ell} \qquad y = \sum_{i \in I} 2^{-i}$$

Tricky reduction:

- Initial points: $(0, 1, I)$ where $I \subseteq \{1, \dots, n\}$

REACH-REGION-TIME is NP-hard (proof idea)

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \ldots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \ldots, n\}, \sum_{i \in I} A_i = B$?

Each point (x, y) encodes a triple (σ, i, I):

$$x = i2^{-k} + \sigma 2^{-\ell} \qquad y = \sum_{i \in I} 2^{-i}$$

Tricky reduction:

- Initial points: $(0, 1, I)$ where $I \subseteq \{1, \ldots, n\}$
- Encoded initial points: $(2^{-k}, q2^{-n})$ where $q \in \{0, \ldots, 2^n - 1\}$
  Not a convex region !

REACH-REGION-TIME is NP-hard (proof idea)

Problem SUBSET-SUM

- **Input:** a goal $B \in \mathbb{N}$ and integers $A_1, \ldots, A_n \in \mathbb{N}$
- **Question:** $\exists I \subseteq \{1, \ldots, n\}, \sum_{i \in I} A_i = B$?

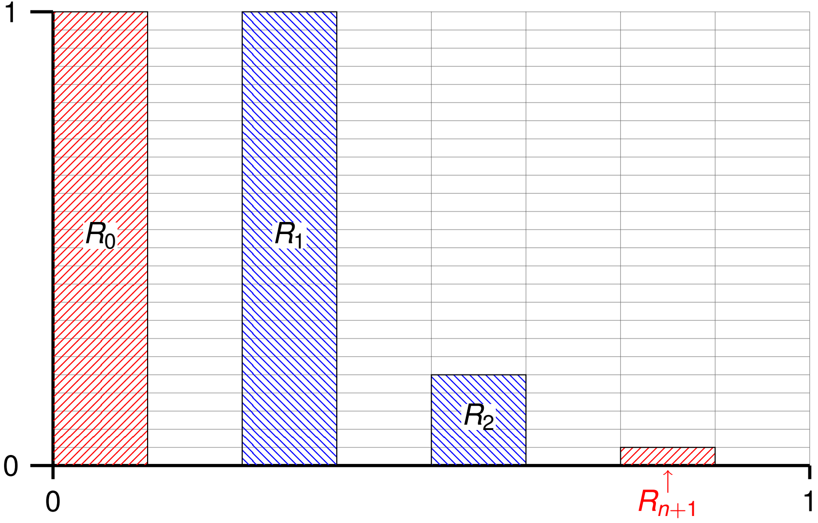
Each point (x, y) encodes a triple (σ, i, I):

$$x = i2^{-k} + \sigma 2^{-\ell} \qquad y = \sum_{i \in I} 2^{-i}$$

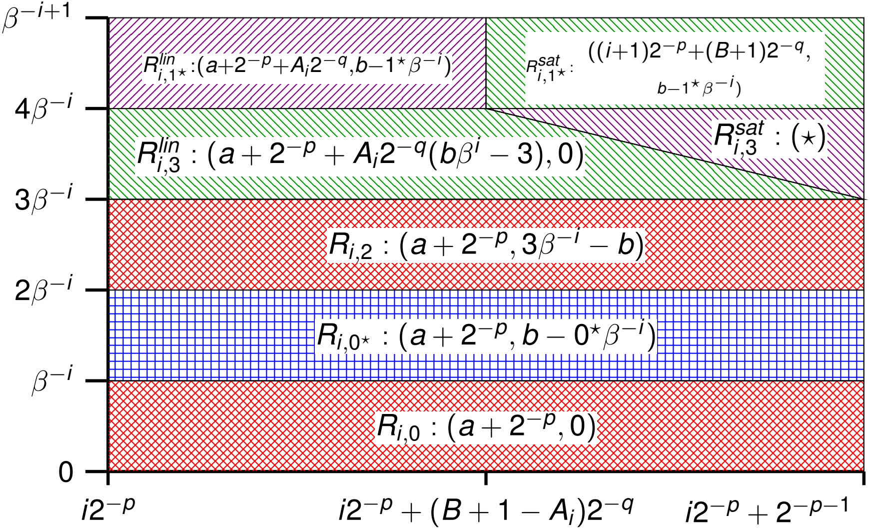
Tricky reduction:

- Initial points: $(0, 1, I)$ where $I \subseteq \{1, \ldots, n\}$
- Encoded initial points: $(2^{-k}, q2^{-n})$ where $q \in \{0, \ldots, 2^n - 1\}$
  Not a convex region !
- Relaxed inital region: $\{2^{-k}\} \times [0, 1]$
  Contains “strange” points

Ok, the actual proof is slightly more complicated...



...horribly more complicated



$(\star)$:$((i+1)2^{-p}+2^{-p-1}-(b\beta^i-3)(2^{-p-1}-(B+1)2^{-q}),0)$

Conclusion

Reachability in piecewise affine systems:

Conclusion

Reachability in piecewise affine systems:

- Undecidable for $d \geqslant 2$

Conclusion

Reachability in piecewise affine systems:

- Undecidable for $d \geqslant 2$
- NP-complete for $d \geqslant 2$ (bounded time variant)

Conclusion

Reachability in piecewise affine systems:

- Undecidable for $d \geqslant 2$
- NP-complete for $d \geqslant 2$ (bounded time variant)
- Open problem for $d = 1$

Conclusion

Reachability in piecewise affine systems:

- Undecidable for $d \geqslant 2$
- NP-complete for $d \geqslant 2$ (bounded time variant)
- Open problem for $d = 1$
- Tricky reduction for the worst case

Conclusion

Reachability in piecewise affine systems:

- Undecidable for $d \geqslant 2$
- NP-complete for $d \geqslant 2$ (bounded time variant)
- Open problem for $d = 1$
- Tricky reduction for the worst case
- Tells us nothing about practical instances

Ordinary Differential Equation

Initial Value Problem

$$y(t_0) = y_0 \qquad y'(t) = f(y(t)) \qquad \forall t \in I$$

where $y : I \to \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^d$

Ordinary Differential Equation

Initial Value Problem

$$y(t_0) = y_0 \qquad y'(t) = f(y(t)) \qquad \forall t \in I$$

where $y : I \to \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^d$

Some problems:

- Compute $y(t) \pm 2^{-n}$ for any $t \in I$ and $n \in \mathbb{N}$

Ordinary Differential Equation

Initial Value Problem

$$y(t_0) = y_0 \qquad y'(t) = f(y(t)) \qquad \forall t \in I$$

where $y : I \to \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^d$

Some problems:

- Compute $y(t) \pm 2^{-n}$ for any $t \in I$ and $n \in \mathbb{N}$
  $\Rightarrow$ Hard even for C^∞ polytime computable f

Ordinary Differential Equation

Initial Value Problem

$$y(t_0) = y_0 \qquad y'(t) = f(y(t)) \qquad \forall t \in I$$

where $y : I \to \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^d$

Some problems:

- Compute $y(t) \pm 2^{-n}$ for any $t \in I$ and $n \in \mathbb{N}$
  $\Rightarrow$ Hard even for C^∞ polytime computable f
- Does $y(t)$ intersect a region R for some $t \in I$? Is I (maximum interval of life) bounded ? Compute I ?

Ordinary Differential Equation

Initial Value Problem

$$y(t_0) = y_0 \qquad y'(t) = f(y(t)) \qquad \forall t \in I$$

where $y : I \to \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^d$

Some problems:

- Compute $y(t) \pm 2^{-n}$ for any $t \in I$ and $n \in \mathbb{N}$
  $\Rightarrow$ Hard even for C^∞ polytime computable f
- Does $y(t)$ intersect a region R for some $t \in I$? Is I (maximum interval of life) bounded ? Compute I ?
- Find attractors. Characterize them. Compute basin of attraction.

Ordinary Differential Equation

Initial Value Problem

$$y(t_0) = y_0 \qquad y'(t) = f(y(t)) \qquad \forall t \in I$$

where $y : I \to \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^d$

Some problems:

- Compute $y(t) \pm 2^{-n}$ for any $t \in I$ and $n \in \mathbb{N}$
  $\Rightarrow$ Hard even for C^∞ polytime computable f
- Does $y(t)$ intersect a region R for some $t \in I$? Is I (maximum interval of life) bounded ? Compute I ?
- Find attractors. Characterize them. Compute basin of attraction. $\Rightarrow$ Usually undecidable even when restricted to computable and C^∞ or analytic f, even in fixed and low (2 or 3) dimension

Ordinary Differential Equation

Initial Value Problem

$$y(t_0) = y_0 \qquad y'(t) = f(y(t)) \qquad \forall t \in I$$

where $y : I \to \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^d$

Some problems:

- Compute $y(t) \pm 2^{-n}$ for any $t \in I$ and $n \in \mathbb{N}$
  $\Rightarrow$ Hard even for C^∞ polytime computable f
- Does $y(t)$ intersect a region R for some $t \in I$? Is I (maximum interval of life) bounded ? Compute I ?
- Find attractors. Characterize them. Compute basin of attraction. $\Rightarrow$ Usually undecidable even when restricted to computable and C^∞ or analytic f, even in fixed and low (2 or 3) dimension

Is there a more restricted and tractable class of ODEs ?

Polynomial Initial Value Problem

Polynomial Initial Value Problem

$$y(t_0) = y_0 \qquad y'(t) = p(y(t)) \qquad \forall t \in I$$

where $y : I \to \mathbb{R}^n$ and p vector of polynomials

Polynomial Initial Value Problem

Polynomial Initial Value Problem

$$y(t_0) = y_0 \qquad y'(t) = p(y(t)) \qquad \forall t \in I$$

where $y : I \to \mathbb{R}^n$ and p vector of polynomials

Example

$$y(0) = 1 \qquad y'(t) = y(t) \quad \rightsquigarrow \quad y(t) = \exp(t)$$

Polynomial Initial Value Problem

Polynomial Initial Value Problem

$$y(t_0) = y_0 \qquad y'(t) = p(y(t)) \qquad \forall t \in I$$

where $y : I \to \mathbb{R}^n$ and p vector of polynomials

Example

$$y(0) = 1 \qquad y'(t) = y(t) \quad \rightsquigarrow \quad y(t) = \exp(t)$$

$$\begin{cases} s(0) = 0 \\ c(0) = 1 \end{cases} \qquad \begin{cases} s'(t) = c(t) \\ c'(t) = -s(t) \end{cases} \quad \rightsquigarrow \quad \begin{cases} s(t) = \sin(t) \\ c(t) = \cos(t) \end{cases}$$

Polynomial Initial Value Problem

Polynomial Initial Value Problem

$$y(t_0) = y_0 \qquad y'(t) = p(y(t)) \qquad \forall t \in I$$

where $y : I \to \mathbb{R}^n$ and p vector of polynomials

Example

$$y(0) = 1 \qquad y'(t) = y(t) \quad \rightsquigarrow \quad y(t) = \exp(t)$$

$$\begin{cases} s(0) = 0 \\ c(0) = 1 \end{cases} \qquad \begin{cases} s'(t) = c(t) \\ c'(t) = -s(t) \end{cases} \quad \rightsquigarrow \quad \begin{cases} s(t) = \sin(t) \\ c(t) = \cos(t) \end{cases}$$

$$y(0) = 1 \qquad y'(t) = -2ty(t)^2 \quad \rightsquigarrow \quad y(t) = \frac{1}{1+t^2}$$

Some problems

- Decide if I (maximum interval of life) is bounded: still undecidable

Some problems

- Decide if I (maximum interval of life) is bounded: still undecidable
- Compute $y(t) \pm 2^{-n}$: P-complete*

Some problems

- Decide if I (maximum interval of life) is bounded: still undecidable
- Compute $y(t) \pm 2^{-n}$: P-complete*

Motivation:

- Simplest nontrivial class: between linear (easy) and analytic (hard)

Some problems

- Decide if I (maximum interval of life) is bounded: still undecidable
- Compute $y(t) \pm 2^{-n}$: P-complete*

Motivation:

- Simplest nontrivial class: between linear (easy) and analytic (hard)
- Captures the General Purpose Analog Computer (GPAC): realistic model of computation
- Contains many interesting systems (most of Newton physics)

Quick recap on GPAC

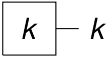
- by Claude Shanon (1941)

Quick recap on GPAC

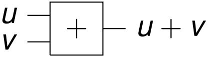
- by Claude Shanon (1941)
- idealization of an analog computer: Differential Analyzer

Quick recap on GPAC

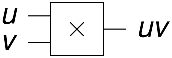
- by Claude Shanon (1941)
- idealization of an analog computer: Differential Analyzer
- circuit built from:



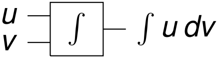
A constant unit



An adder unit



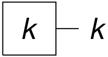
An multiplier unit



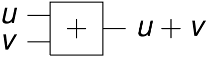
An integrator unit

Quick recap on GPAC

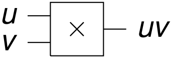
- by Claude Shanon (1941)
- idealization of an analog computer: Differential Analyzer
- circuit built from:



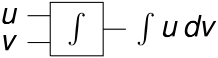
A constant unit



An adder unit



An multiplier unit



An integrator unit

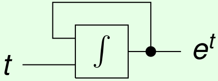
Theorem

y is generated by a GPAC iff it is a component of a PIVP

It exists !

GPAC: examples

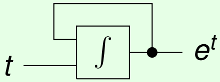
Example (One variable, linear system)



$$\begin{cases} y' = y \\ y(0) = 1 \end{cases}$$

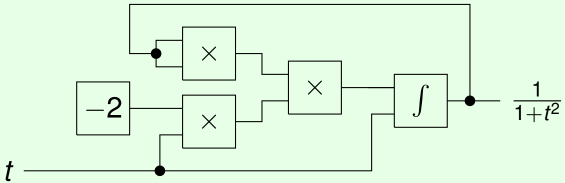
GPAC: examples

Example (One variable, linear system)



$$\begin{cases} y' = y \\ y(0) = 1 \end{cases}$$

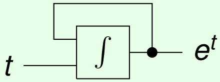
Example (One variable, nonlinear system)



$$\begin{cases} y' = -2ty^2 \\ y(0) = 1 \end{cases}$$

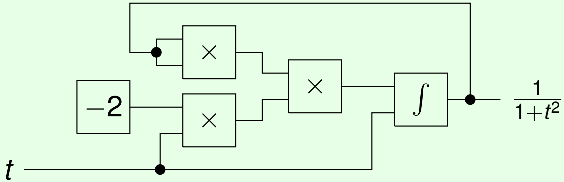
GPAC: examples

Example (One variable, linear system)



$$\begin{cases} y' = y \\ y(0) = 1 \end{cases}$$

Example (Two variable, nonlinear system)



$$\begin{cases} y' = -2ty^2 \\ y(0) = 1 \\ t' = 1 \\ t(0) = 0 \end{cases}$$

Solving PIVP over unbounded domain

Assume that $y : I \to \mathbb{R}^d$ satisfies:

$$y(t_0) = y_0 \qquad y'(t) = p(y(t)) \quad \forall t \in I$$

Solving PIVP over unbounded domain

Assume that $y : I \to \mathbb{R}^d$ satisfies:

$$y(t_0) = y_0 \qquad y'(t) = p(y(t)) \quad \forall t \in I$$

Theorem (Previously known)

If I is compact then y is polynomial time computable over I:

Solving PIVP over unbounded domain

Assume that $y : I \to \mathbb{R}^d$ satisfies:

$$y(t_0) = y_0 \qquad y'(t) = p(y(t)) \quad \forall t \in I$$

Theorem (Previously known)

If I is compact then y is polynomial time computable over I: computing $y(t) \pm 2^{-n}$ takes time poly(n) (n unary)

Solving PIVP over unbounded domain

Assume that $y : I \to \mathbb{R}^d$ satisfies:

$$y(t_0) = y_0 \qquad y'(t) = p(y(t)) \quad \forall t \in I$$

Theorem (Previously known)

If I is compact then y is polynomial time computable over I: computing $y(t) \pm 2^{-n}$ takes time poly(n) (n unary)

Issues with this:

- Doesn't work for unbounded I

Solving PIVP over unbounded domain

Assume that $y : I \to \mathbb{R}^d$ satisfies:

$$y(t_0) = y_0 \qquad y'(t) = p(y(t)) \quad \forall t \in I$$

Theorem (Previously known)

If I is compact then y is polynomial time computable over I: computing $y(t) \pm 2^{-n}$ takes time poly(n) (n unary)

Issues with this:

- Doesn't work for unbounded I
- How does the complexity depend on y_0, d, p, I ?

Solving PIVP over unbounded domain (cont.)

Theorem (Our work)

Computing $y(t) \pm 2^{-n}$ takes time:

$$\text{poly}(\deg p, \log \|y_0\|, \log \Sigma p, n, \ell(t_0, t))^d$$

where:

Solving PIVP over unbounded domain (cont.)

Theorem (Our work)

Computing $y(t) \pm 2^{-n}$ takes time:

$$\text{poly}(\deg p, \log \|y_0\|, \log \Sigma p, n, \ell(t_0, t))^d$$

where:

- Σp: sum of absolute value of coefficients of p

Solving PIVP over unbounded domain (cont.)

Theorem (Our work)

Computing $y(t) \pm 2^{-n}$ takes time:

$$\text{poly}(\deg p, \log \|y_0\|, \log \Sigma p, n, \ell(t_0, t))^d$$

where:

- Σp: sum of absolute value of coefficients of p
- $\ell(t_0, t)$: “length” of y over $[t_0, t]$

Solving PIVP over unbounded domain (cont.)

Theorem (Our work)

Computing $y(t) \pm 2^{-n}$ takes time:

$$\text{poly}(\deg p, \log \|y_0\|, \log \Sigma p, n, \ell(t_0, t))^d$$

where:

- Σp: sum of absolute value of coefficients of p
- $\ell(t_0, t)$: "length" of y over $[t_0, t]$

$$\ell(t_0, t) = \int_{t_0}^{t} \max(1, \|y'(u)\|) du$$

Solving PIVP over unbounded domain (cont.)

Theorem (Our work)

Computing $y(t) \pm 2^{-n}$ takes time:

$$\text{poly}(\deg p, \log \|y_0\|, \log \Sigma p, n, \ell(t_0, t))^d$$

where:

- Σp: sum of absolute value of coefficients of p
- $\ell(t_0, t)$: "length" of y over $[t_0, t]$

$$\ell(t_0, t) = \int_{t_0}^{t} \max(1, \|y'(u)\|) du$$

Notes:

- also works if p has PTIME computable coefficients (like π)

Solving PIVP over unbounded domain (cont.)

Theorem (Our work)

Computing $y(t) \pm 2^{-n}$ takes time:

$$\text{poly}(\deg p, \log \|y_0\|, \log \Sigma p, n, \ell(t_0, t))^d$$

where:

- Σp: sum of absolute value of coefficients of p
- $\ell(t_0, t)$: "length" of y over $[t_0, t]$

$$\ell(t_0, t) = \int_{t_0}^{t} \max(1, \|y'(u)\|) du$$

Notes:

- also works if p has PTIME computable coefficients (like π)
- the algorithm can find $\ell(t_0, t)$ automagically

Proof details

- Standard numerical analysis technique: Taylor series

Proof details

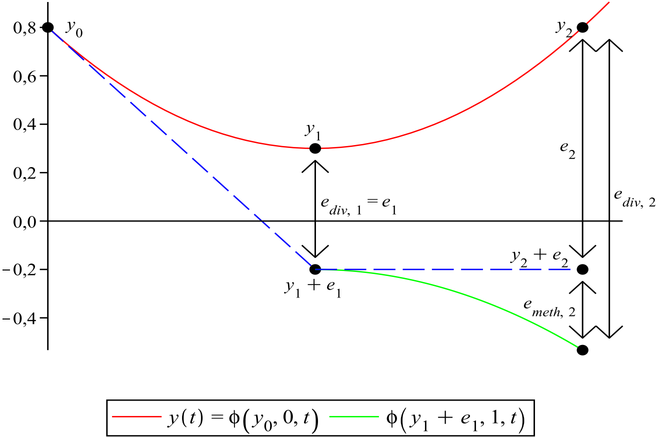
- Standard numerical analysis technique: Taylor series
- Throw in adaptive steps, variable order

Proof details

- Standard numerical analysis technique: Taylor series
- Throw in adaptive steps, variable order
- Mix with incremental refinement, stop criterion and tricky analysis

Proof details

- Standard numerical analysis technique: Taylor series
- Throw in adaptive steps, variable order
- Mix with incremental refinement, stop criterion and tricky analysis



Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}}\varepsilon^{-\frac{1}{\omega-1}}\right)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|)du$$

Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}}\varepsilon^{-\frac{1}{\omega-1}}\right)$
Euler ($\omega = 2$)	1	$\mathcal{O}\left(\frac{L^3}{\varepsilon}\right)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|)du$$

Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}}\varepsilon^{-\frac{1}{\omega-1}}\right)$
Euler ($\omega = 2$)	1	$\mathcal{O}\left(\frac{L^3}{\varepsilon}\right)$
Taylor2 ($\omega = 3$)	2	$\mathcal{O}\left(\frac{L^2}{\sqrt{\varepsilon}}\right)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|)du$$

Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}}\varepsilon^{-\frac{1}{\omega-1}}\right)$
Euler ($\omega = 2$)	1	$\mathcal{O}\left(\frac{L^3}{\varepsilon}\right)$
Taylor2 ($\omega = 3$)	2	$\mathcal{O}\left(\frac{L^2}{\sqrt{\varepsilon}}\right)$
Taylor4 ($\omega = 5$)	4	$\mathcal{O}\left(\frac{L^{3/2}}{\sqrt[4]{\varepsilon}}\right)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|)du$$

Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}}\varepsilon^{-\frac{1}{\omega-1}}\right)$
Euler ($\omega = 2$)	1	$\mathcal{O}\left(\frac{L^3}{\varepsilon}\right)$
Taylor2 ($\omega = 3$)	2	$\mathcal{O}\left(\frac{L^2}{\sqrt{\varepsilon}}\right)$
Taylor4 ($\omega = 5$)	4	$\mathcal{O}\left(\frac{L^{3/2}}{\sqrt[4]{\varepsilon}}\right)$
Smart $\left(\omega = 1 + \log\frac{L}{\varepsilon}\right)$	$\log\frac{L}{\varepsilon}$	$\mathcal{O}\left(L^{\sim 1}\right)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|)du$$

Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}}\varepsilon^{-\frac{1}{\omega-1}}\right)$
Euler ($\omega = 2$)	1	$\mathcal{O}\left(\frac{L^3}{\varepsilon}\right)$
Taylor2 ($\omega = 3$)	2	$\mathcal{O}\left(\frac{L^2}{\sqrt{\varepsilon}}\right)$
Taylor4 ($\omega = 5$)	4	$\mathcal{O}\left(\frac{L^{3/2}}{\sqrt[4]{\varepsilon}}\right)$
Smart $\left(\omega = 1 + \log \frac{L}{\varepsilon}\right)$	$\log \frac{L}{\varepsilon}$	$\mathcal{O}\left(L^{\sim 1}\right)$
Taylor∞ ($\omega = \infty$)	∞	$\mathcal{O}(L)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|)du$$

Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}}\varepsilon^{-\frac{1}{\omega-1}}\right)$
Euler ($\omega = 2$)	1	$\mathcal{O}\left(\frac{L^3}{\varepsilon}\right)$
Taylor2 ($\omega = 3$)	2	$\mathcal{O}\left(\frac{L^2}{\sqrt{\varepsilon}}\right)$
Taylor4 ($\omega = 5$)	4	$\mathcal{O}\left(\frac{L^{3/2}}{\sqrt[4]{\varepsilon}}\right)$
Smart $\left(\omega = 1 + \log \frac{L}{\varepsilon}\right)$	$\log \frac{L}{\varepsilon}$	$\mathcal{O}\left(L^{\sim 1}\right)$
Taylor∞ ($\omega = \infty$)	∞	$\mathcal{O}(L)$
Variable	$\mathcal{O}\left(\log \frac{L}{\varepsilon}\right)$	$\mathcal{O}(L)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|)du$$

Characterization of Turing polynomial time

Definition: $\mathcal{L} \subseteq \{0,1\}^*$ is polytime-recognizable iff for all w:

Characterization of Turing polynomial time

Definition: $\mathcal{L} \subseteq \{0,1\}^*$ is polytime-recognizable iff for all w:

$$y(0) = q(\psi(w)) \qquad y' = p(y) \qquad \psi(w) = \sum_{i=1}^{|w|} w_i 2^{-i}$$

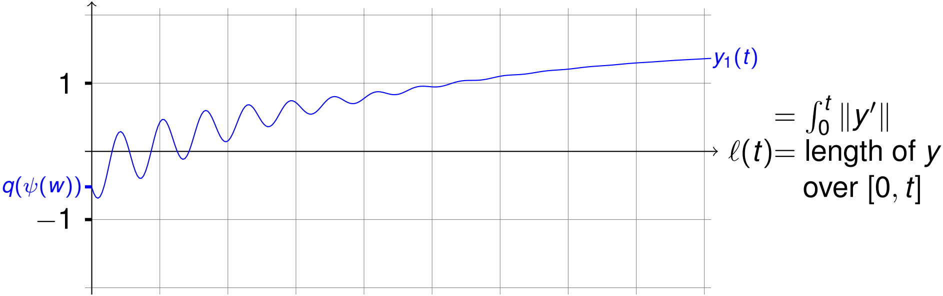
satisfies:

Characterization of Turing polynomial time

Definition: $\mathcal{L} \subseteq \{0,1\}^*$ is polytime-recognizable iff for all w:

$$y(0) = q(\psi(w)) \qquad y' = p(y) \qquad \psi(w) = \sum_{i=1}^{|w|} w_i 2^{-i}$$

satisfies:

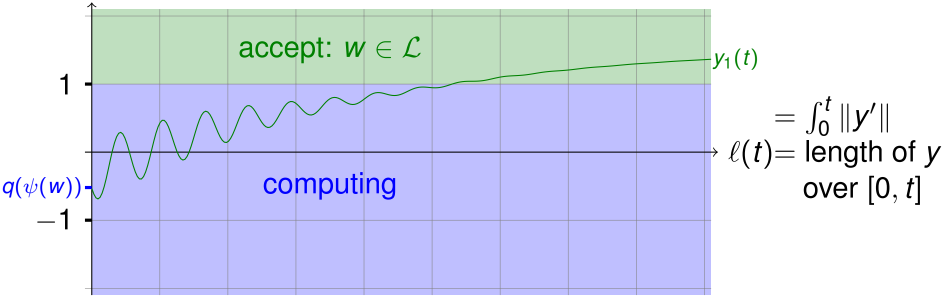


Characterization of Turing polynomial time

Definition: $\mathcal{L} \subseteq \{0,1\}^*$ is polytime-recognizable iff for all w:

$$y(0) = q(\psi(w)) \qquad y' = p(y) \qquad \psi(w) = \sum_{i=1}^{|w|} w_i 2^{-i}$$

satisfies:



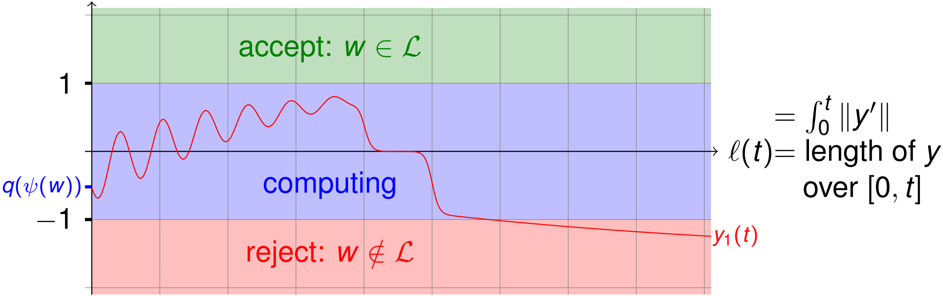
1. if $y_1(t) \geqslant 1$ then $w \in \mathcal{L}$

Characterization of Turing polynomial time

Definition: $\mathcal{L} \subseteq \{0,1\}^*$ is polytime-recognizable iff for all w:

$$y(0) = q(\psi(w)) \qquad y' = p(y) \qquad \psi(w) = \sum_{i=1}^{|w|} w_i 2^{-i}$$

satisfies:



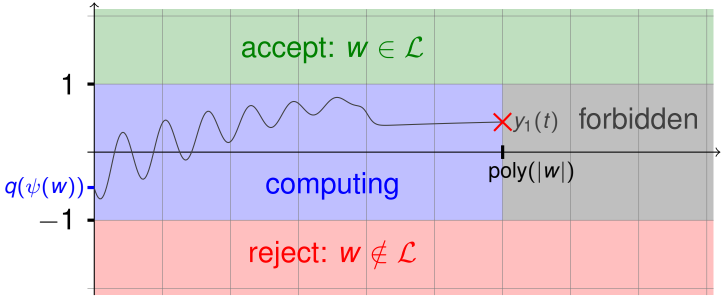
2. if $y_1(t) \leqslant -1$ then $w \notin \mathcal{L}$

Characterization of Turing polynomial time

Definition: $\mathcal{L} \subseteq \{0,1\}^*$ is polytime-recognizable iff for all w:

$$y(0) = q(\psi(w)) \qquad y' = p(y) \qquad \psi(w) = \sum_{i=1}^{|w|} w_i 2^{-i}$$

satisfies:



$\ell(t)$= length of y over $[0,t]$ $= \int_0^t \|y'\|$

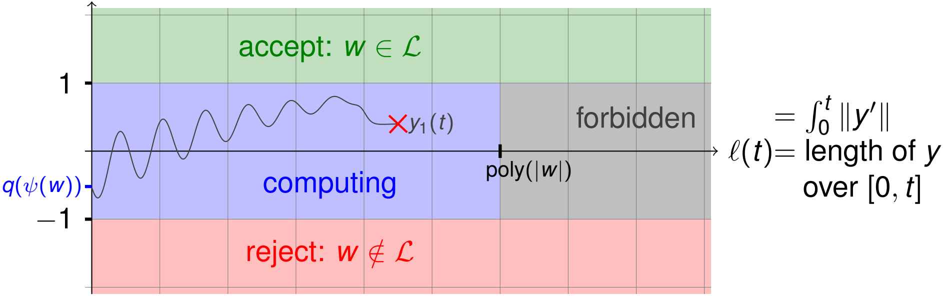
3. if $\ell(t) \geqslant \text{poly}(|w|)$ then $|y_1(t)| \geqslant 1$

Characterization of Turing polynomial time

Definition: $\mathcal{L} \subseteq \{0,1\}^*$ is polytime-recognizable iff for all w:

$$y(0) = q(\psi(w)) \qquad y' = p(y) \qquad \psi(w) = \sum_{i=1}^{|w|} w_i 2^{-i}$$

satisfies:



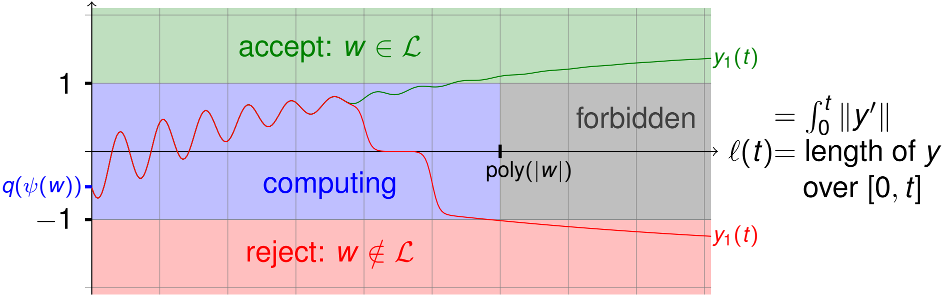
4. $\ell(t) \geqslant t$

Characterization of Turing polynomial time

Definition: $\mathcal{L} \subseteq \{0,1\}^*$ is polytime-recognizable iff for all w:

$$y(0) = q(\psi(w)) \qquad y' = p(y) \qquad \psi(w) = \sum_{i=1}^{|w|} w_i 2^{-i}$$

satisfies:



Theorem

$\mathcal{L} \in$ P if and only if $\mathcal{L}$ is polytime-recognizable.

PIVP hardness

Corollary

The following problem is *P*-complete:
Input: $y_0 \in \mathbb{Q}^d$, $\Upsilon \in \mathbb{N}$ in unary, p polynomial

PIVP hardness

Corollary

The following problem is *P*-complete:
Input: $y_0 \in \mathbb{Q}^d$, $\Upsilon \in \mathbb{N}$ in unary, p polynomial
Assumption: $\exists y : [0,1] \to \mathbb{R}^d$ s.t. $y(0) = y_0$, $y' = p(y)$ and $\|y(t)\| \leqslant \Upsilon$

PIVP hardness

Corollary

The following problem is *P*-complete:
Input: $y_0 \in \mathbb{Q}^d$, $\Upsilon \in \mathbb{N}$ in unary, p polynomial
Assumption: $\exists y : [0,1] \to \mathbb{R}^d$ s.t. $y(0) = y_0$, $y' = p(y)$ and $\|y(t)\| \leqslant \Upsilon$
Assumption: $y(1) \geqslant 1$ or $y(1) \leqslant -1$

PIVP hardness

Corollary

The following problem is *P*-complete:

Input: $y_0 \in \mathbb{Q}^d$, $\Upsilon \in \mathbb{N}$ in unary, p polynomial

Assumption: $\exists y : [0,1] \to \mathbb{R}^d$ s.t. $y(0) = y_0$, $y' = p(y)$ and $\|y(t)\| \leqslant \Upsilon$

Assumption: $y(1) \geqslant 1$ or $y(1) \leqslant -1$

Decide: $y(1) \geqslant 1$?

Proof gem: iteration with differential equations

Goal

Iterate $\tilde{f}$ with a PIVP: $y(n) \approx \tilde{f}^{[n]}([x])$

Proof gem: iteration with differential equations

Goal

Iterate $\tilde{f}$ with a PIVP: $y(n) \approx \tilde{f}^{[n]}([x])$

Idea (dimension one): $\tilde{f} : \mathbb{R} \to \mathbb{R}$

- Use two variables: y and z, two phases: $[0, 1/2]$ and $[1/2, 1]$.

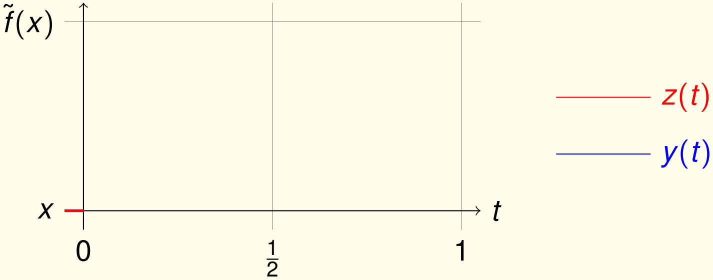
Proof gem: iteration with differential equations

Goal

Iterate $\tilde{f}$ with a PIVP: $y(n) \approx \tilde{f}^{[n]}([x])$

Idea (dimension one): $\tilde{f} : \mathbb{R} \to \mathbb{R}$

- Use two variables: y and z, two phases: $[0, 1/2]$ and $[1/2, 1]$.



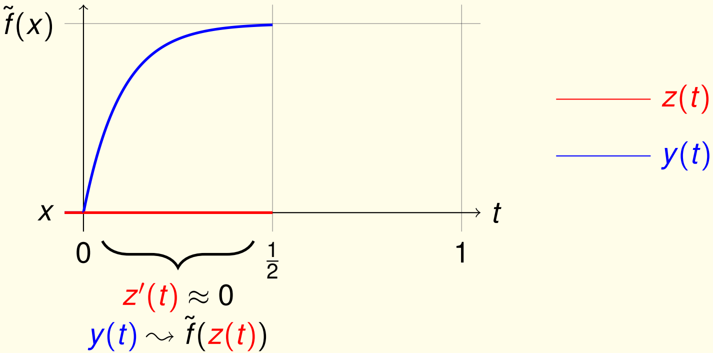
Proof gem: iteration with differential equations

Goal

Iterate $\tilde{f}$ with a PIVP: $y(n) \approx \tilde{f}^{[n]}([x])$

Idea (dimension one): $\tilde{f} : \mathbb{R} \to \mathbb{R}$

- Use two variables: y and z, two phases: $[0, 1/2]$ and $[1/2, 1]$.



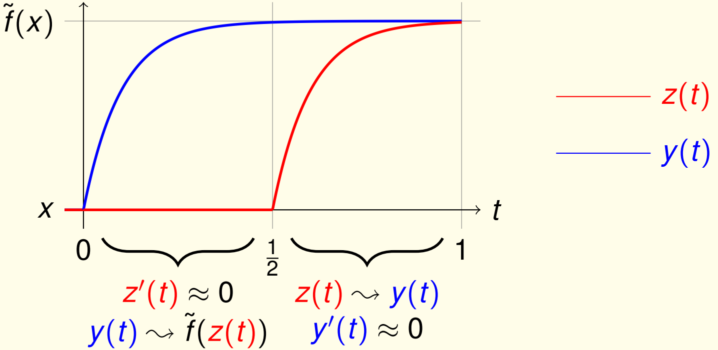
Proof gem: iteration with differential equations

Goal

Iterate $\tilde{f}$ with a PIVP: $y(n) \approx \tilde{f}^{[n]}([x])$

Idea (dimension one): $\tilde{f}: \mathbb{R} \to \mathbb{R}$

- Use two variables: y and z, two phases: $[0, 1/2]$ and $[1/2, 1]$.



Conclusion

- Polynomial ODEs: good compromise between power and tractability
- Point to region reachability with condition: P-complete

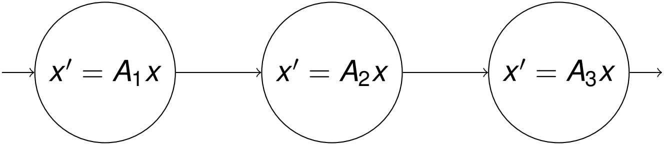
Conclusion

- Polynomial ODEs: good compromise between power and tractability
- Point to region reachability with condition: P-complete

Future work:

- Extend PIVP solving to more general ODEs
- More general reachability
- Study practical complexity of PIVP solving
- Other measures of complexity
- More efficient algorithm for systems with more properties ?

Linear Hybrid Automata



- Finite number of control states
- Each state has a linear continuous dynamic: $x' = Ax$
- Nondeterministic transitions between states (no guards)

Small math recap

The solution to a linear system of differential equations

$$x(0) = x_0, \qquad x'(t) = Ax(t)$$

Small math recap

The solution to a linear system of differential equations

$$x(0) = x_0, \qquad x'(t) = Ax(t)$$

is given by:

$$x(t) = x_0 e^{At}$$

where the exponential of matrices is given by

$$\exp(M) = \sum_{n=0}^{\infty} \frac{M^n}{n!}.$$

Problems

Reachability:

- Point to point (Orbit problem)
- Point to region (Hitting problem)
- Escape problem

Problems

Reachability:

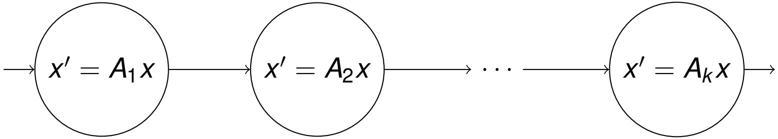
- Point to point (Orbit problem)
- Point to region (Hitting problem)
- Escape problem

Transformation:

- What kind of transformations can be achieved ?

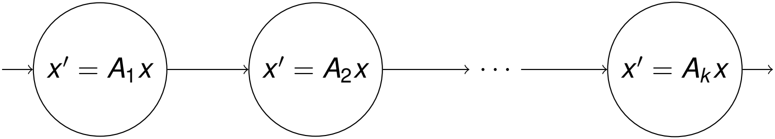
Matrix-Exponential Problems

Given *algebraic* matrices $A_1, \ldots, A_k, C$.



Matrix-Exponential Problems

Given *algebraic* matrices $A_1, \ldots, A_k, C$.



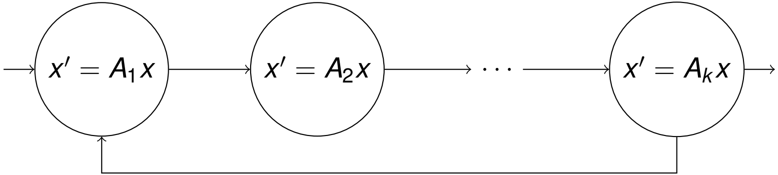
Definition (Matrix-Exponential Problem)

Decide if there exists $t_1, \ldots, t_k \geqslant 0$ such that:

$$\prod_{i=1}^{k} e^{A_i t_i} = C.$$

Matrix-Exponential Semigroup Prolem

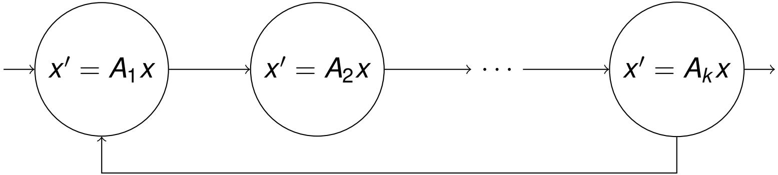
Given *algebraic* matrices $A_1, \ldots, A_k, C$.



NOTE: equivalent to a complete graph

Matrix-Exponential Semigroup Prolem

Given *algebraic* matrices $A_1, \ldots, A_k, C$.



NOTE: equivalent to a complete graph

Definition (Matrix-Exponential Semigroup Problem)

Decide if C belongs to the semigroup generated by:

$$\{\exp(A_i t) : t \geqslant 0, i = 1, \ldots, k\}.$$

Results:

Theorem (Commutative case)

If the matrices $A_1, \ldots, A_k$ commute, Matrix-Exponential and Matrix-Exponential Semigroup problems are equivalent and decidable.

Results:

Theorem (Commutative case)

If the matrices $A_1, \ldots, A_k$ commute, Matrix-Exponential and Matrix-Exponential Semigroup problems are equivalent and decidable.

Theorem (General case)

In general, Matrix-Exponential and Matrix-Exponential Semigroup problems are undecidable.

Questions ?

- Do you have any questions ?