

Computability and complexity over the reals: some characterisations of classes

Manon Blanc

IRN CLove



université
PARIS-SACLAY



Introduction

- We are interested in having *algebraic characterisations* of PTIME and PSPACE.

→ Why algebraic characterisations?

- We are interested in having *algebraic characterisations* of PTIME and PSPACE.

→ Why algebraic characterisations?

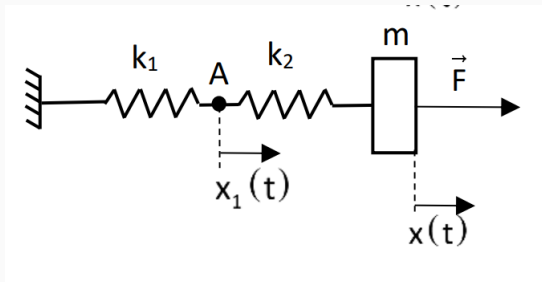
- We already do it for computability classes

- We are interested in having *algebraic characterisations* of PTIME and PSPACE.

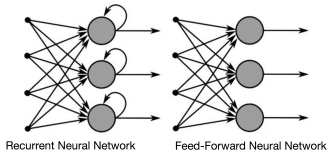
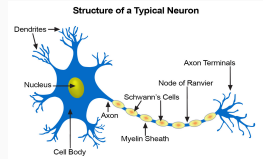
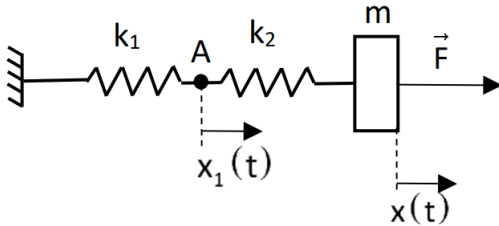
→ Why algebraic characterisations?

- We already do it for computability classes
- Computation over the reals : Framework of computable analysis

Introduction



Introduction



- Framework of **implicit complexity** (Cobham, Bellantoni, Cook, Levant, Marion, de Naurois...)
- **Time = Length:**
 - Continuous ODEs : Bournez, Graça, Pouly (ICALP 2017)
 - Discrete ODEs : Bournez, Durand (MFCS 2019)

**A type of ODE for algebraic
characterisations: PTIME**

Discrete derivation

$$\frac{\delta f(\mathbf{x}, \mathbf{y})}{\delta \mathbf{x}} = f(\mathbf{x} + 1, \mathbf{y}) - f(\mathbf{x}, \mathbf{y})$$

→ It is a derivation with a step of one

“Length ODE”?

Definition (Length ODE)

A function \mathbf{f} is “length-ODE” definable (from u , g and h) if it is a solution of:

$$f(0, \mathbf{y}) = \mathbf{g}(\mathbf{y}) \quad \text{and} \quad \frac{\delta \mathbf{f}(x, \mathbf{y})}{\delta \ell} = \mathbf{u}(\mathbf{f}(x, \mathbf{y}), \mathbf{h}(x, \mathbf{y}), x, \mathbf{y}). \quad (1)$$

“Length ODE”?

Definition (Length ODE)

A function \mathbf{f} is “length-ODE” definable (from u , g and h) if it is a solution of:

$$f(0, \mathbf{y}) = \mathbf{g}(\mathbf{y}) \quad \text{and} \quad \frac{\delta \mathbf{f}(x, \mathbf{y})}{\delta \ell} = \mathbf{u}(\mathbf{f}(x, \mathbf{y}), \mathbf{h}(x, \mathbf{y}), x, \mathbf{y}). \quad (1)$$

Formal synonym for right-hand side of (1):

$$\mathbf{f}(x + 1, \mathbf{y}) = \mathbf{f}(x, \mathbf{y}) + (\ell(x + 1) - \ell(x)) \cdot \mathbf{u}(\mathbf{f}(x, \mathbf{y}), \mathbf{h}(x, \mathbf{y}), x, \mathbf{y})$$

“Length ODE”?

$$\mathbf{f}(x + 1, \mathbf{y}) = \mathbf{f}(x, \mathbf{y}) + (\ell(x + 1) - \ell(x)) \cdot \mathbf{u}(\mathbf{f}(x, \mathbf{y}), \mathbf{h}(x, \mathbf{y}), x, \mathbf{y})$$

Derivation with respect to the length = change of variable.

Variation when:

$$\ell(x + 1) - \ell(x) \neq 0$$

Inspired by:

$$\frac{\delta f(x, \mathbf{y})}{\delta x} = \frac{\delta \ell(x)}{\delta x} \cdot \frac{\delta f(x, \mathbf{y})}{\delta \ell(x)}.$$

Motivation behind Length-ODEs



Fig. 5. A continuous system before and after an exponential speed-up.

θ solution of:

$$y' = f(y)$$

$$f : \mathbb{R} \rightarrow \mathbb{R}$$

ϕ solution of:

$$z = z'$$

$$y' = f(y)z$$

Motivation behind Length-ODEs



Fig. 5. A continuous system before and after an exponential speed-up.

$$\text{Re-scaling: } \phi_1(t) = \theta(e^t)$$

Motivation behind Length-ODEs

Now “time-complexity” is measured by the **length** of the solution curve of the ODE.

Invariance by rescaling

Example of length-ODE

$$f(0) = 2$$

$$\frac{\delta f}{\delta \ell}(x) = f(x) \cdot f(x) - f(x)$$

Example of length-ODE

$$f(0) = 2$$

$$\frac{\delta f}{\delta \ell}(x) = f(x) \cdot f(x) - f(x)$$

Unique solution : $f(x) = 2^{2^{\ell(x)}}$

Example of length-ODE

$$f(0) = 2; \quad \frac{\delta f}{\delta \ell}(x) = f(x) \cdot f(x) - f(x)$$

Example of length-ODE

$$f(0) = 2; \quad \frac{\delta f}{\delta \ell}(x) = f(x) \cdot f(x) - f(x)$$
$$f(x+1) - f(x) = (\ell(x+1) - \ell(x))(f(x)f(x) - f(x))$$

Example of length-ODE

$$f(0) = 2; \quad \frac{\delta f}{\delta \ell}(x) = f(x) \cdot f(x) - f(x)$$
$$f(x+1) - f(x) = (\ell(x+1) - \ell(x))(f(x)f(x) - f(x))$$

$$z = \ell(x)$$

$$F(z) = 2^{2^z}$$

Example of length-ODE

$$f(0) = 2; \quad \frac{\delta f}{\delta \ell}(x) = f(x) \cdot f(x) - f(x)$$
$$f(x+1) - f(x) = (\ell(x+1) - \ell(x))(f(x)f(x) - f(x))$$

$$z = \ell(x)$$

$$F(z) = 2^{2^z}$$

- We have $f(x) = F(z)$

Example of length-ODE

$$f(0) = 2; \quad \frac{\delta f}{\delta \ell}(x) = f(x) \cdot f(x) - f(x)$$
$$f(x+1) - f(x) = (\ell(x+1) - \ell(x))(f(x)f(x) - f(x))$$

$$z = \ell(x)$$

$$F(z) = 2^{2^z}$$

- We have $f(x) = F(z)$
- $F(z+1) = 2^{2^z+2^z} = F(z)F(z)$

Example of length-ODE

$$f(0) = 2; \quad \frac{\delta f}{\delta \ell}(x) = f(x) \cdot f(x) - f(x)$$
$$f(x+1) - f(x) = (\ell(x+1) - \ell(x))(f(x)f(x) - f(x))$$

$$z = \ell(x)$$

$$F(z) = 2^{2^z}$$

- We have $f(x) = F(z)$
- $F(z+1) = 2^{2^z+2^z} = F(z)F(z)$
- Then, $F(z+1) = F(z) + (z+1 - z)(F(z)F(z) - F(z))$

u is essentially linear in f iff:

$$u = A(\dots, \sigma(f)) \cdot f(x) + B(\dots, \sigma(f))$$

with A and B that may depend on a sigmoid over f .

Characterisation of PTIME (the one you are used to)

We consider

$$\text{LDL} = [0, 1, \pi_i^k, \ell(x), +, -, \times, \text{cond}(x);$$

composition, linear length ODE]

Theorem (Bournez & Durand, '19)

$$\text{LDL} \cap \mathbb{N}^{\mathbb{N}} = \text{PTIME} \cap \mathbb{N}^{\mathbb{N}}$$

Some basic definitions of computable analysis

Definitions (from computable analysis): intuition

- What is a **computable real number** x ?
 - we can compute a representation of x .

Definitions (from computable analysis): intuition

- What is a **computable real number** x ?
→ we can compute a representation of x .

Example

e, π are computable. $\sum_{i \geq 1} 2^{-BB(i)}$, where BB is the Busy Beavers function is not.

Definitions (from computable analysis): intuition

- What is a **computable real number** x ?
→ we can compute a representation of x .

Example

e, π are computable. $\sum_{i \geq 1} 2^{-BB(i)}$, where BB is the Busy Beavers function is not.

- What is a **computable function** $f : \mathbb{R} \rightarrow \mathbb{R}$?
→ on a representation of $x \in \mathbb{R}$, we produce a representation of $f(x)$.

Definitions (from computable analysis): intuition

- What is a **computable real number** x ?
→ we can compute a representation of x .

Example

e, π are computable. $\sum_{i \geq 1} 2^{-BB(i)}$, where BB is the Busy Beavers function is not.

- What is a **computable function** $f : \mathbb{R} \rightarrow \mathbb{R}$?
→ on a representation of $x \in \mathbb{R}$, we produce a representation of $f(x)$.
- How do we define **complexity over the reals**?

More formally

- What is a **computable real number** x ?

There exists a Cauchy sequence $\phi : \mathbb{N} \rightarrow \mathbb{D}$ such that for all $n \in \mathbb{N}$, $|\phi(n) - x| \leq 2^{-n}$.

More formally

- What is a **computable real number** x ?

There exists a Cauchy sequence $\phi : \mathbb{N} \rightarrow \mathbb{D}$ such that for all $n \in \mathbb{N}$, $|\phi(n) - x| \leq 2^{-n}$.

- What is a **computable function** $f : \mathbb{R} \rightarrow \mathbb{R}$?

There exists an oracle Turing machine M such that, on a Cauchy sequence ϕ converging to x , M queries the oracle to have m such that $|\phi(m) - x| \leq 2^{-m}$ and computes $d \in \mathbb{D}$ such that $|d - f(x)| \leq 2^{-n}$.

PTIME over the reals with **D**iscrete **O**DEs

First step: real sequences

Definition

$f : \mathbb{R} \rightarrow \mathbb{R}$ is an **effective limit** of $\bar{f} : \mathbb{R} \times \mathbb{N} \rightarrow \mathbb{R}$ if

$$|f(x) - \bar{f}(x, 2^n)| \leq 2^{-n}$$

Key observation : \bar{f} computable in polynomial time $\Rightarrow f$ computable in polynomial time.

Algebraic characterisation of PTIME for sequences

We consider:

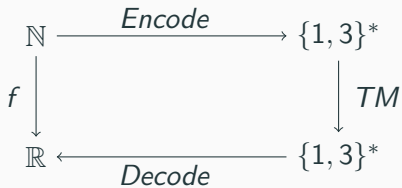
$$\overline{\text{LDL}}^\bullet = [\mathbf{0}, \mathbf{1}, \pi_i^k, \ell(x), +, -, \times, \overline{\text{cond}}(x), \frac{x}{2};$$

composition, linear length ODE, effective limit]

Theorem (B., Bournez, MCU22)

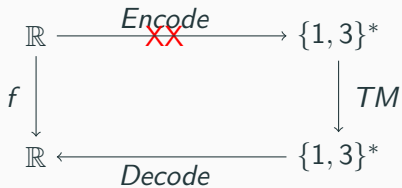
$$\text{FP}\text{TIME} \cap \mathbb{R}^{\mathbb{N}} = \overline{\text{LDL}}^\bullet \cap \mathbb{R}^{\mathbb{N}}$$

“Proof”



Second step: functions over the reals with discrete ODEs

What we cannot do



FPTIME for computable functions over the reals

We consider:

$$\overline{\text{LDL}}^\circ = [\mathbf{0}, \mathbf{1}, \pi_i^k, \ell(x), +, -, \tanh, \frac{x}{2}, \frac{x}{3};$$

composition, linear length ODE, effective limit]

Theorem (B., Bournez, MFCS23)

$$\overline{\text{LDL}}^\circ \cap \mathbb{R}^{\mathbb{R}} = \text{FPTIME} \cap \mathbb{R}^{\mathbb{R}}$$

Proof ideas (\supseteq)

- We give *continuous approximations* of some basic functions (integer/fractional part, modulo 2, ...);

Proof ideas (\supseteq)

- We give *continuous approximations* of some basic functions (integer/fractional part, modulo 2, ...);
- We *encode each step of the transition of the Turing machine* into the algebra, using the previous functions : Next;

Proof ideas (\supseteq)

- We give *continuous approximations* of some basic functions (integer/fractional part, modulo 2, ...);
- We *encode each step of the transition of the Turing machine* into the algebra, using the previous functions : Next;
- We obtain a *linear length ODE* encoding the execution of the Turing machine: $State(t + 1) = Next(State(t))$ so

$$\frac{\delta \overline{State}(2^{t+1}, y)}{\delta \ell} = \overline{Next}(\overline{State}(2^t, y))$$

PSPACE with Discrete ODEs

Definition (Robust linear ODE)

A bounded function \mathbf{f} is a robustly linear ODE definable from g and u (with \mathbf{u} *essentially linear* in $\mathbf{f}(x, \mathbf{y})$) if:

1. it is a solution of

$$\mathbf{f}(0, \mathbf{y}) = \mathbf{g}(\mathbf{y}) \quad \text{and} \quad \frac{\delta \mathbf{f}(x, \mathbf{y})}{\delta x} = \mathbf{u}(\mathbf{f}(x, \mathbf{y}), \mathbf{h}(x, \mathbf{y}), x, \mathbf{y}),$$

2. the ODE is (polynomially) **numerically stable**.

FPSPACE for computable functions over the reals

We consider:

$$\overline{\text{RLD}}^\circ = [\mathbf{0}, \mathbf{1}, \pi_i^k, \ell(n), +, -, \tanh, \frac{x}{2}, \frac{x}{3};$$

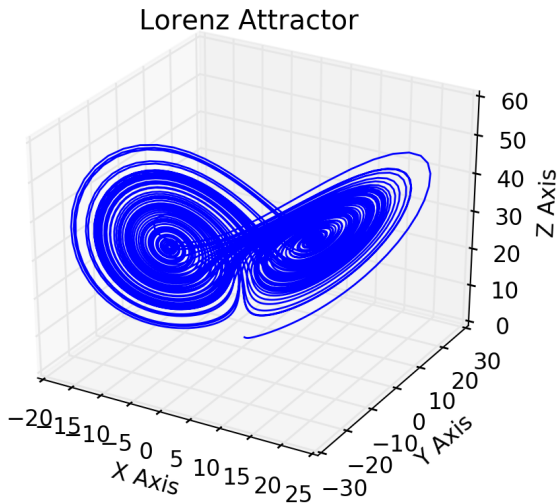
composition, robust linear ODE, ELim]

Theorem (B., Bournez, MFCS23)

$$\overline{\text{RLD}}^\circ \cap \mathbb{R}^{\mathbb{R}} = \text{FPSPACE} \cap \mathbb{R}^{\mathbb{R}}$$

Functions over the reals with continuous ODEs

Why it is not easy to solve ODEs



Why it is not easy to solve ODEs

$$\mathbf{f}(0, \mathbf{x}) = \mathbf{g}(\mathbf{x}) \quad \text{and} \quad \frac{\partial \mathbf{f}(t, \mathbf{x})}{\partial t} = \mathbf{u}(\mathbf{f}(t, \mathbf{x}), t, \mathbf{x}),$$

- Non-computable in the **general case**, for \mathbf{u} computable (Pour-El, Richards)

Why it is not easy to solve ODEs

$$\mathbf{f}(0, \mathbf{x}) = \mathbf{g}(\mathbf{x}) \quad \text{and} \quad \frac{\partial \mathbf{f}(t, \mathbf{x})}{\partial t} = \mathbf{u}(\mathbf{f}(t, \mathbf{x}), t, \mathbf{x}),$$

- Non-computable in the **general case**, for \mathbf{u} computable (Pour-El, Richards)
- Computable for \mathbf{u} **computable** and unique solution (Collins, Graça, Ruohonen)

Why it is not easy to solve ODEs

$$\mathbf{f}(0, \mathbf{x}) = \mathbf{g}(\mathbf{x}) \quad \text{and} \quad \frac{\partial \mathbf{f}(t, \mathbf{x})}{\partial t} = \mathbf{u}(\mathbf{f}(t, \mathbf{x}), t, \mathbf{x}),$$

- Non-computable in the **general case**, for \mathbf{u} computable (Pour-El, Richards)
- Computable for \mathbf{u} **computable** and unique solution (Collins, Graça, Ruohonen)
- PSPACE-completeness on a **bounded domain** for \mathbf{u} computable in polynomial-time (Kawamura, Ko)

Robust Continuous ODEs (1/2)

A function $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$ is robustly ODE definable (from initial condition \mathbf{g} , and dynamic \mathbf{u}) if

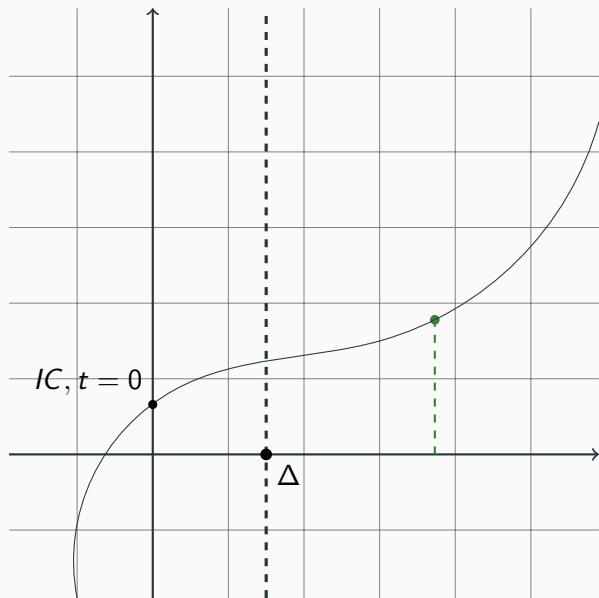
- it corresponds to the solution of the following continuous ODE:

$$\mathbf{f}(0, \mathbf{x}) = \mathbf{g}(\mathbf{x}) \quad \text{and} \quad \frac{\partial \mathbf{f}(t, \mathbf{x})}{\partial t} = \mathbf{u}(\mathbf{f}(t, \mathbf{x}), t, \mathbf{x}), \quad (2)$$

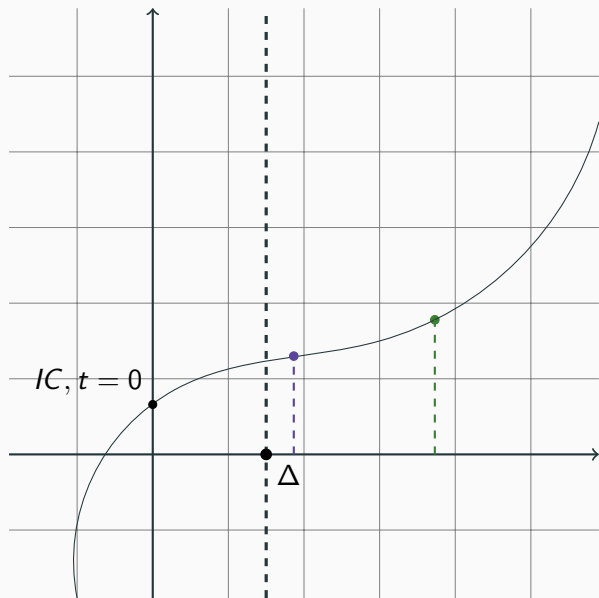
- And polynomially numerically stable (to be defined : next slides)

→ this allows computation by dichotomy.

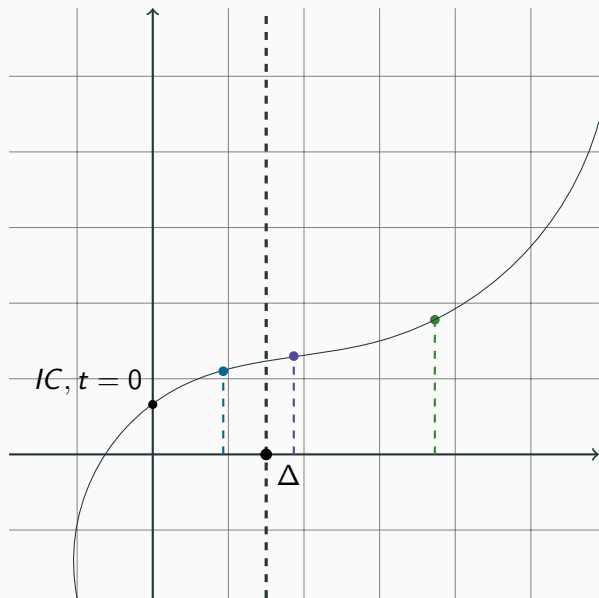
An unusual way of solving an ODE



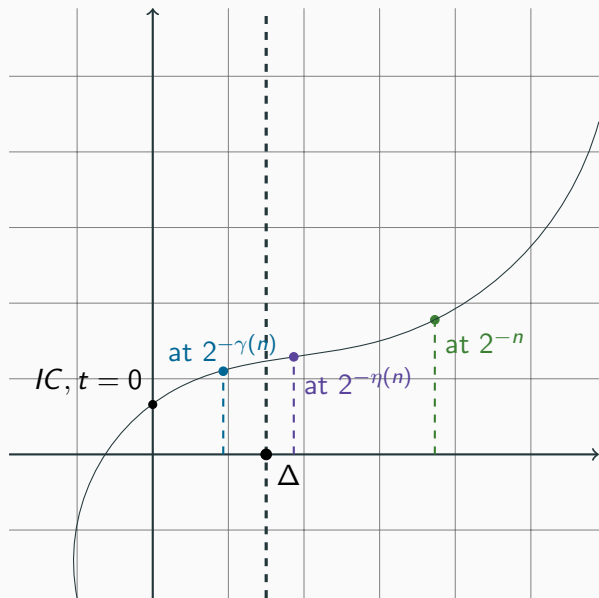
An unusual way of solving an ODE



An unusual way of solving an ODE



An unusual way of solving an ODE



An unusual way of solving an ODE

- Computation at $t = x$ at precision 2^{-n} , with initial condition $t = 0$
- Computation at $t = t_0 + \frac{x}{2}$ at precision $2^{-\eta(n)}$, with initial condition $t_0 = 0$ and $t_0 = \frac{x}{2}$
- Computation at $t = t_0 + \frac{x}{4}$ at precision $2^{-\gamma(n)}$, with initial condition $t_0 = 0$ and $t_0 = \frac{x}{4}$ and $t_0 = \frac{x}{2}$ and $t_0 = \frac{3x}{4}$

Robust Continuous ODEs (2/2)

More “formally” :

- $\exists \Delta \in \mathbb{Q}_+^*$, such that the previous ODE is (polynomially spaced) **solvable** on $[0, \Delta]$.
- For $t \geq \Delta$, we can compute $\mathbf{f}(t, \mathbf{x})$ at 2^{-n} by computing some approximation $\widetilde{\mathbf{f}(t/2, \mathbf{x})}$ of $\mathbf{f}(t/2, \mathbf{x})$ at precision $2^{-\eta(n)}$.

Characterisation of PSPACE with continuous ODEs

We consider :

$$\overline{\text{RCD}} = [\mathbf{0}, \mathbf{1}, \pi_i^k, +, -, \times, \tanh, \cos, \pi, \frac{\times}{2}, \frac{\times}{3};$$

composition, robust continuous ODE, ELim]

Theorem (B., Bournez, ICALP24)

$$\overline{\text{RCD}} \cap \mathbb{R}^{\mathbb{R}} = \text{FPSPACE} \cap \mathbb{R}^{\mathbb{R}}$$

$$\overline{\text{RLD}^\circ} \cap \mathbb{R}^{\mathbb{R}} = \overline{\text{RCD}} \cap \mathbb{R}^{\mathbb{R}}$$

$$\overline{\text{RLD}^\circ} \cap \mathbb{R}^{\mathbb{R}} = \overline{\text{RCD}} \cap \mathbb{R}^{\mathbb{R}}$$

- **Discrete** \Rightarrow **Continuous settings:**

Adaptation of the a trick due to Branicky ('95)

$$\overline{\text{RLD}^\circ} \cap \mathbb{R}^{\mathbb{R}} = \overline{\text{RCD}} \cap \mathbb{R}^{\mathbb{R}}$$

- **Discrete** \Rightarrow **Continuous settings:**

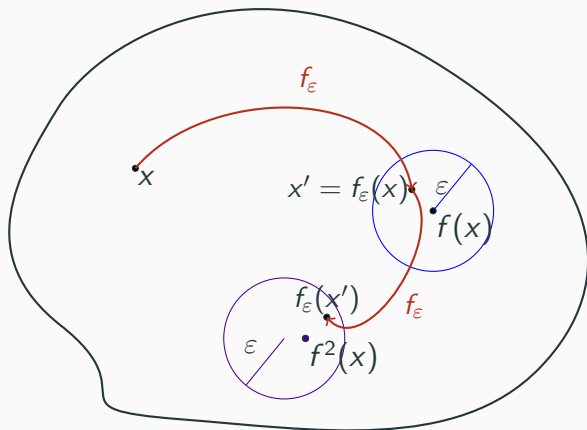
Adaptation of the a trick due to Branicky ('95)

- **Continuous** \Rightarrow **Discrete settings:**

From our definitions of robustness of ODEs

Further work

PSPACE-completeness: a bit on robust dynamical systems



- Reachability in dynamical systems is *undecidable*
- Reachability in **robust** dynamical systems is *decidable* (Asarin & Bouajjani)

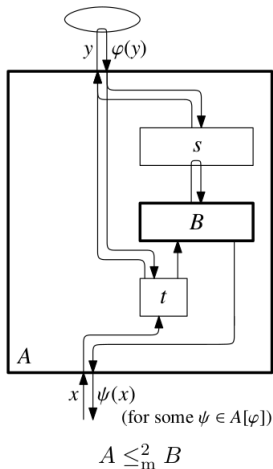
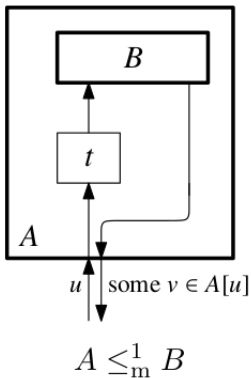
Theorem (B., Bournez, CSL24)

Take a locally Lipschitz system, with $f : X \rightarrow X$ polynomial-time computable, with X a closed rational box. Then, for $p : \mathbb{N} \rightarrow \mathbb{N}$ a polynomial, $R_p^{\mathcal{P}} \subseteq \mathbb{Q}^d \times \mathbb{Q}^d \times \mathbb{N} \in \text{PSPACE}$.

→ works also for dynamical systems over the reals

PSPACE-completeness: Type-2 reduction

We take the reduction function function as defined by Cook and Kawamura (the drawing comes from their paper):



Prove that reachability is PSPACE-hard for robust dynamical systems:

- We use the algebra $\overline{\text{LDL}^\circ}$ and $\overline{\text{RLD}^\circ}$;
- We use the Type-2 reduction.

Some more further works

- talking about NP (non-determinism)?
- talking about probabilistic classes
- talking about distributions

I'm looking for a post-doc (starting September 2025)