Verification of Hybrid Systems

Béatrice Bérard

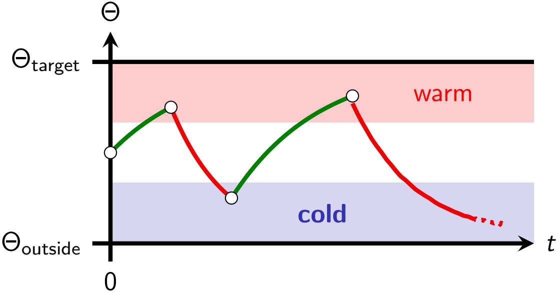
Sorbonne Université –LIP6

Based on joint work with:
P. Bouyer, S. Haddad, V. Jugé, C. Picaronny, M. Safey El Din, M. Sassolas

GALA, December 14th, 2019

Hybrid systems

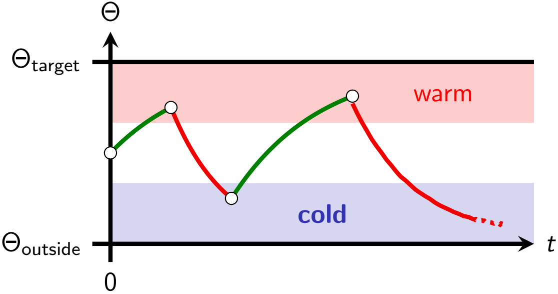
Hybrid systems



Two modes:

1. Heater **ON**: $\dot{\Theta} = \alpha(\Theta_{\text{target}} - \Theta)$
2. Heater **OFF**: $\dot{\Theta} = \beta(\Theta_{\text{outside}} - \Theta)$

Hybrid systems

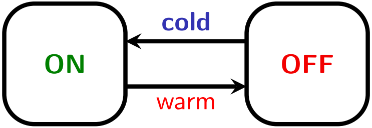


Two modes:

1. Heater **ON**: $\dot{\Theta} = \alpha(\Theta_{\text{target}} - \Theta)$
2. Heater **OFF**: $\dot{\Theta} = \beta(\Theta_{\text{outside}} - \Theta)$

Duality between:

- **Discrete** set of system modes
- **Continuous** system evolution



Thanks to V. Jugé

Verification

Verification problems are mostly undecidable on hybrid systems

Decidability requires restricting:

- either the flows [Henzinger et al. 1998]

  for instance with clocks: $\dot{x} = 1$ in all modes

- or the jumps [Alur et al. 2000]

  using for instance strong resets between modes

Other approaches

like

- bounded delay reachability,
- or approximations by discrete transition systems.

Outline

Timed Automata from Alur, Dill (1990)

Polynomial Interrupt Timed Automata

Reachability using cylindrical decomposition

Algorithmic issues

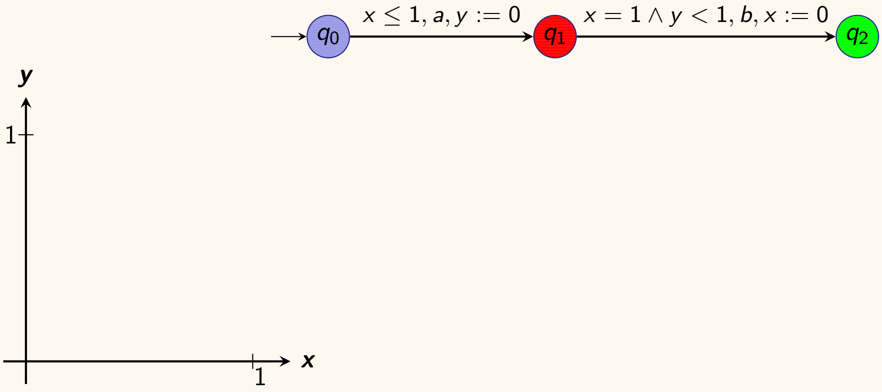
A result on Dynamical Systems

Timed automata

Variables: clocks with flow $\dot{x} = 1$ for each $x \in X$
Guards: conjunctions of $x \bowtie k$, with $k \in \mathbb{N}$ and $\bowtie$ in $\{<, \leq, =, \geq, >\}$
Updates: conjunctions of reset $x := 0$

Clock valuation: $v = (v(x_1), \ldots, v(x_n)) \in \mathbb{R}_+^n$ if $X = \{x_1, \ldots, x_n\}$

A geometric view of a trajectory



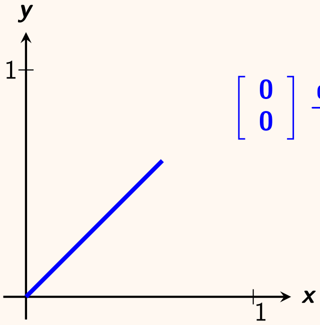
Timed automata

Variables: clocks with flow $\dot{x} = 1$ for each $x \in X$
Guards: conjunctions of $x \bowtie k$, with $k \in \mathbb{N}$ and $\bowtie$ in $\{<, \leq, =, \geq, >\}$
Updates: conjunctions of reset $x := 0$

Clock valuation: $v = (v(x_1), \ldots, v(x_n)) \in \mathbb{R}_+^n$ if $X = \{x_1, \ldots, x_n\}$

A geometric view of a trajectory

$$\to q_0 \xrightarrow{x \leq 1, a, y := 0} q_1 \xrightarrow{x = 1 \wedge y < 1, b, x := 0} q_2$$



$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \xrightarrow{0.6} \begin{bmatrix} 0.6 \\ 0.6 \end{bmatrix}$$

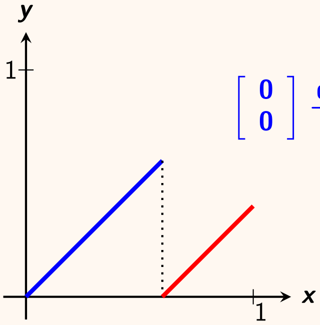
Timed automata

Variables: clocks with flow $\dot{x} = 1$ for each $x \in X$
Guards: conjunctions of $x \bowtie k$, with $k \in \mathbb{N}$ and $\bowtie$ in $\{<, \leq, =, \geq, >\}$
Updates: conjunctions of reset $x := 0$

Clock valuation: $v = (v(x_1), \ldots, v(x_n)) \in \mathbb{R}^n_+$ if $X = \{x_1, \ldots, x_n\}$

A geometric view of a trajectory

$$\rightarrow q_0 \xrightarrow{x \leq 1, a, y := 0} q_1 \xrightarrow{x = 1 \wedge y < 1, b, x := 0} q_2$$



$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \xrightarrow{0.6} \begin{bmatrix} 0.6 \\ 0.6 \end{bmatrix} \xrightarrow{a} \begin{bmatrix} 0.6 \\ 0 \end{bmatrix} \xrightarrow{0.4} \begin{bmatrix} 1 \\ 0.4 \end{bmatrix}$$

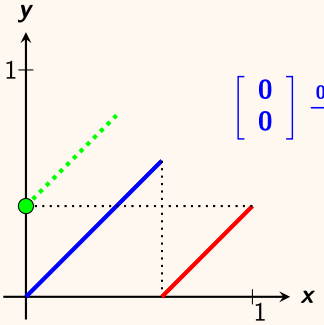
Timed automata

Variables: clocks with flow $\dot{x} = 1$ for each $x \in X$
Guards: conjunctions of $x \bowtie k$, with $k \in \mathbb{N}$ and $\bowtie$ in $\{<, \leq, =, \geq, >\}$
Updates: conjunctions of reset $x := 0$

Clock valuation: $v = (v(x_1), \ldots, v(x_n)) \in \mathbb{R}_+^n$ if $X = \{x_1, \ldots, x_n\}$

A geometric view of a trajectory

$$\rightarrow q_0 \xrightarrow{x \leq 1, a, y := 0} q_1 \xrightarrow{x = 1 \wedge y < 1, b, x := 0} q_2$$



$$\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \xrightarrow{0.6} \begin{bmatrix} \mathbf{0.6} \\ \mathbf{0.6} \end{bmatrix} \xrightarrow{a} \begin{bmatrix} \mathbf{0.6} \\ \mathbf{0} \end{bmatrix} \xrightarrow{0.4} \begin{bmatrix} \mathbf{1} \\ \mathbf{0.4} \end{bmatrix} \xrightarrow{b} \begin{bmatrix} \mathbf{0} \\ \mathbf{0.4} \end{bmatrix}$$

Reachability

Semantics of $\mathcal{A}$

with clocks $X = \{x_1, \dots, x_n\}$, set of modes Q, set of transitions E:
a transition system $\mathcal{T}_{\mathcal{A}}$ with

- **configurations:** $(q, v) \in Q \times \mathbb{R}_+^n$
- **time steps:** $(q, v) \xrightarrow{d} (q, v + d)$
- **discrete steps:** $(q, v) \xrightarrow{e} (q', v')$ for a transition $e = q \xrightarrow{g,a,r} q'$ in E if clock values v satisfy the guard g and $v' = v[r]$

An execution is a sequence alternating time and discrete steps.

Reachability problem

Given $\mathcal{A}$ and $q_f \in Q$
is there an execution from initial configuration $s_0 = (q_0, \mathbf{0})$ to (q_f, v)
for some valuation v ?

A finite quotient for timed automata

[Alur, Dill, 1990]

From $\mathcal{A}$, build a finite automaton $Reg(\mathcal{A})$ preserving reachability.

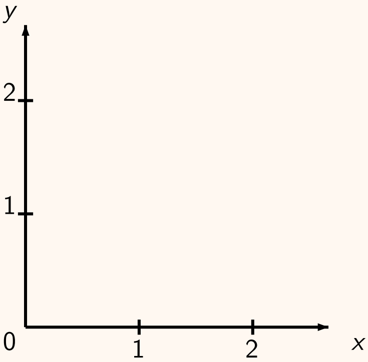
Equivalence $\sim$ over $\mathbb{R}_+^n$ producing a partition $\mathcal{R}$ of regions

The automaton $Reg(\mathcal{A})$ is time-abstract bisimilar to $\mathcal{T}_{\mathcal{A}}$:

- set of states $Q \times \mathcal{R}$,
- abstract time steps $(q, R) \to (q, succ(R))$ consistent with time elapsing in $\mathcal{T}_{\mathcal{A}}$,
- discrete steps $(q, R) \xrightarrow{e} (q', R')$ consistent with discrete transitions in $\mathcal{T}_{\mathcal{A}}$.

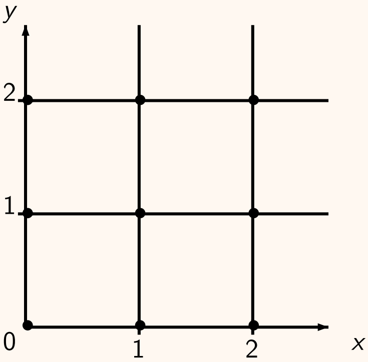
Quotient construction

A geometric view with two clocks x and y, maximal constant $m = 2$



Quotient construction

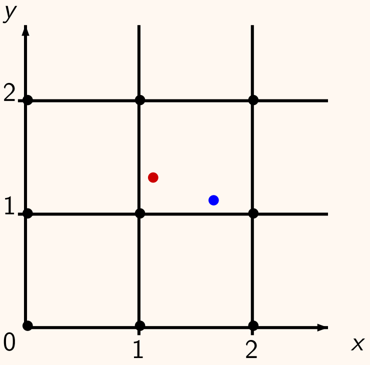
A geometric view with two clocks x and y, maximal constant $m = 2$



- Equivalent valuations must be consistent with constraints $x \bowtie k$

Quotient construction

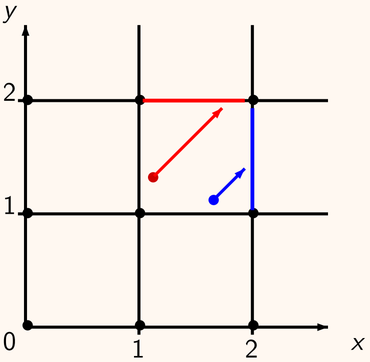
A geometric view with two clocks x and y, maximal constant $m = 2$



- Equivalent valuations must be consistent with constraints $x \bowtie k$
- Equivalent valuations must be consistent with time elapsing

Quotient construction

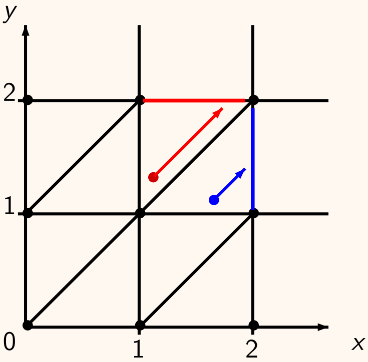
A geometric view with two clocks x and y, maximal constant $m = 2$



- Equivalent valuations must be consistent with constraints $x \bowtie k$
- Equivalent valuations must be consistent with time elapsing

Quotient construction

A geometric view with two clocks x and y, maximal constant $m = 2$

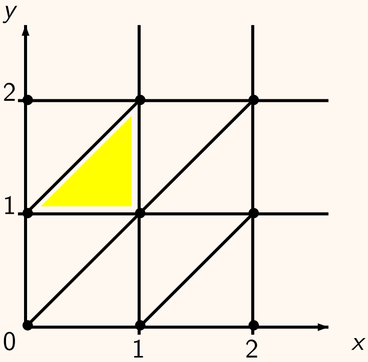


number of regions in $O(|X|! \cdot m^{|X|})$

- Equivalent valuations must be consistent with constraints $x \bowtie k$
- Equivalent valuations must be consistent with time elapsing

Quotient construction

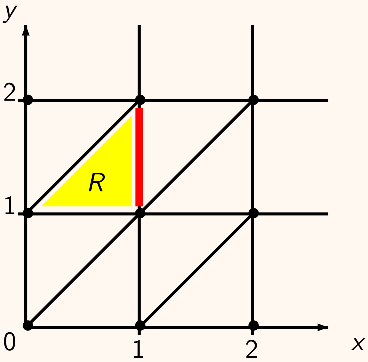
A geometric view with two clocks x and y, maximal constant $m = 2$



region R defined by
$0 < x < 1$ and $1 < y < 2$
and $y < x + 1$

Quotient construction

A geometric view with two clocks x and y, maximal constant $m = 2$

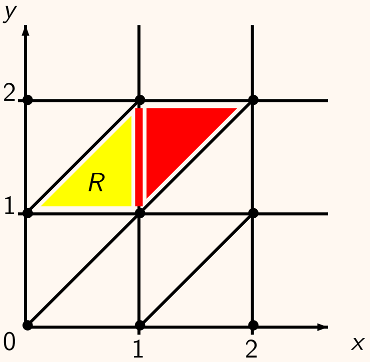


region R defined by
$0 < x < 1$ and $1 < y < 2$
and $y < x + 1$

Time successor of R
$x = 1$ and $1 < y < 2$

Quotient construction

A geometric view with two clocks x and y, maximal constant $m = 2$

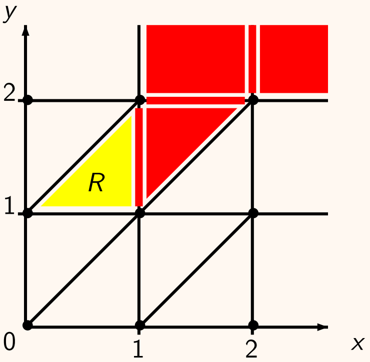


region R defined by
$0 < x < 1$ and $1 < y < 2$
and $y < x + 1$

Time successor of R
$x = 1$ and $1 < y < 2$

Quotient construction

A geometric view with two clocks x and y, maximal constant $m = 2$

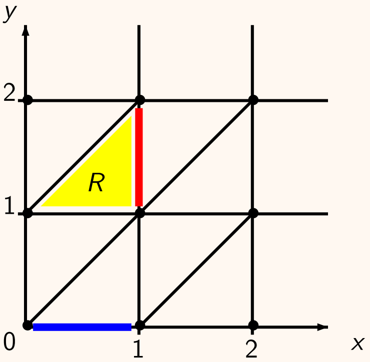


region R defined by
$0 < x < 1$ and $1 < y < 2$
and $y < x + 1$

Time successor of R
$x = 1$ and $1 < y < 2$

Quotient construction

A geometric view with two clocks x and y, maximal constant $m = 2$

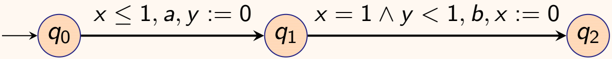


region R defined by
$0 < x < 1$ and $1 < y < 2$
and $y < x + 1$

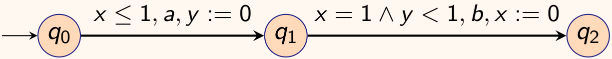
Time successor of R
$x = 1$ and $1 < y < 2$

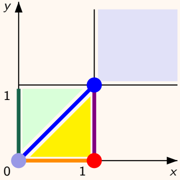
Discrete step from R
with $y := 0$
$0 < x < 1$ and $y = 0$

Example of quotient

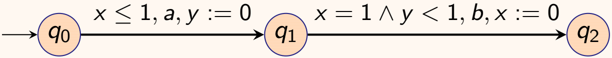


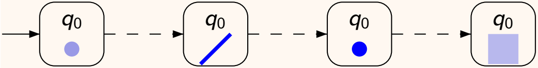
Example of quotient

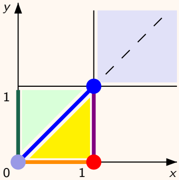




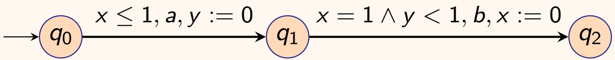
Example of quotient

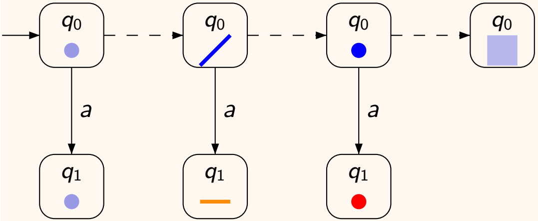


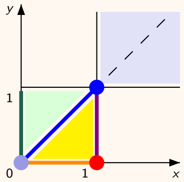




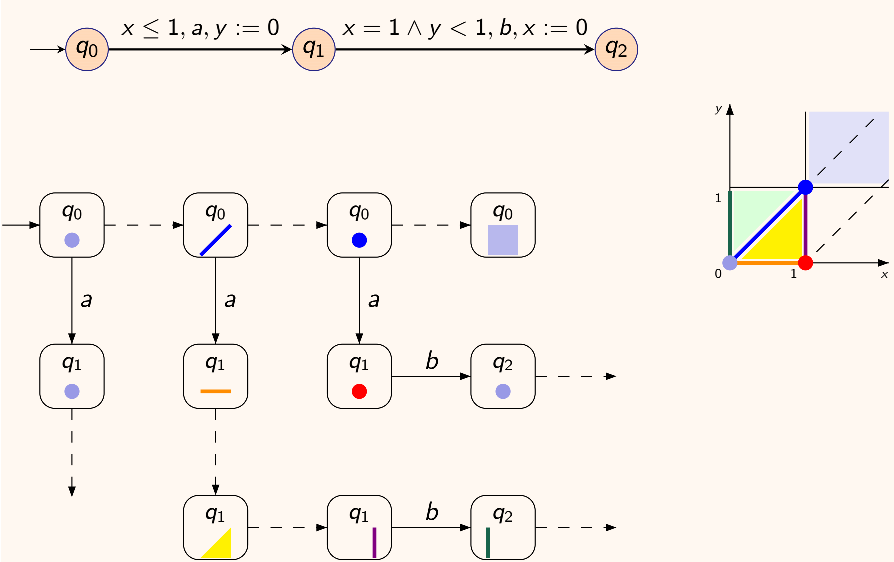
Example of quotient







Example of quotient



Outline

Timed Automata from Alur, Dill (1990)

Polynomial Interrupt Timed Automata

Reachability using cylindrical decomposition

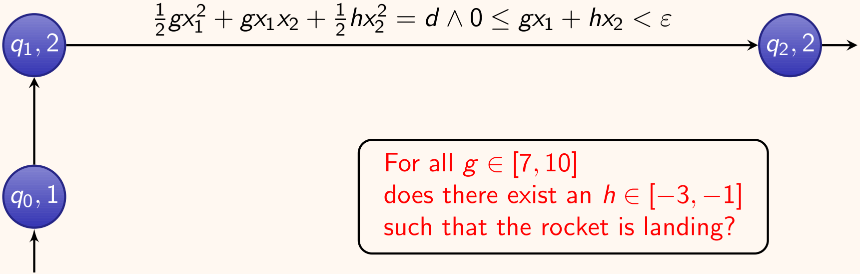
Algorithmic issues

A result on Dynamical Systems

Polynomial constraints with parameters

Landing a rocket

- First stage (lasting x_1) in state q_0:
  From distance d, the rocket approaches the land under gravitation g;
- Second stage (lasting x_2, while x_1 is frozen) in q_1:
  The rocket approaches the land with constant deceleration $h < 0$;
- Third stage: The rocket must reach the land
  with small positive speed (less than ε).

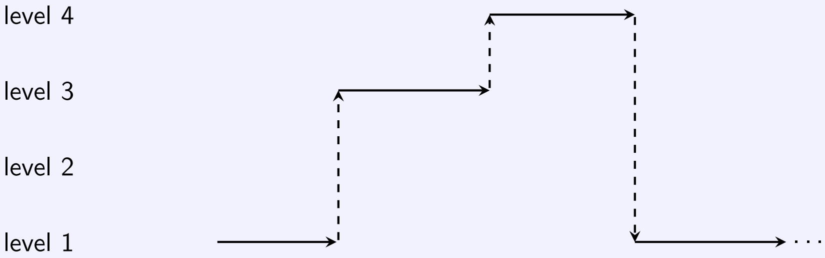


For all $g \in [7, 10]$
does there exist an $h \in [-3, -1]$
such that the rocket is landing?

Interrupt clocks

Many real-time systems include interruption mechanisms (as in processors).

Several levels with exactly one active clock at each level

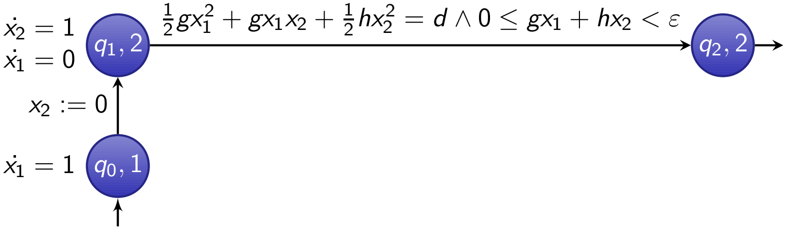


$$\begin{bmatrix} x_4 \\ x_3 \\ x_2 \\ x_1 \end{bmatrix} \quad \text{Exec:} \quad \begin{bmatrix} {\color{red}0} \\ {\color{red}0} \\ {\color{red}0} \\ 0 \end{bmatrix} \xrightarrow{1.5} \begin{bmatrix} {\color{red}0} \\ 0 \\ 0 \\ 1.5 \end{bmatrix} \xrightarrow{2.1} \begin{bmatrix} 0 \\ 2.1 \\ 0 \\ 1.5 \end{bmatrix} \xrightarrow{1.7} \begin{bmatrix} {\color{red}0} \\ {\color{red}0} \\ {\color{red}0} \\ 1.5 \end{bmatrix} \xrightarrow{2.2} \begin{bmatrix} {\color{red}0} \\ {\color{red}0} \\ {\color{red}0} \\ 3.7 \end{bmatrix}$$

Polynomial Interrupt Timed Automata

In the class PolITA

- variables are interrupt clocks with flow $\dot{x} = 0$ or $\dot{x} = 1$
  ordered along hierarchical levels,
- guards are polynomial constraints
  and variables can be updated by polynomials.



Main result: Reachability is decidable in 2EXPTIME

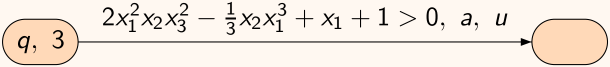
[BHPSS 15]

PolITA: Syntax

clocks $X = \{x_1, \ldots, x_n\}$ with x_k active at level k,
set of modes Q with $\lambda : Q \to \{1, \ldots, n\}$ the state level,
Guards: conjunctions of polynomial constraints
$P \bowtie 0$ with $\bowtie$ in $\{<, \leq, =, \geq, >\}$, and $P \in \mathbb{Q}[x_1, \ldots, x_k]$ at level k.

PolITA: Syntax

clocks $X = \{x_1, \ldots, x_n\}$ with x_k active at level k,
set of modes Q with $\lambda : Q \to \{1, \ldots, n\}$ the state level,
Guards: conjunctions of polynomial constraints
$P \bowtie 0$ with $\bowtie$ in $\{<, \leq, =, \geq, >\}$, and $P \in \mathbb{Q}[x_1, \ldots, x_k]$ at level k.



PolITA: Syntax

clocks $X = \{x_1, \ldots, x_n\}$ with x_k active at level k,
set of modes Q with $\lambda : Q \to \{1, \ldots, n\}$ the state level,
Guards: conjunctions of polynomial constraints
$P \bowtie 0$ with $\bowtie$ in $\{<, \leq, =, \geq, >\}$, and $P \in \mathbb{Q}[x_1, \ldots, x_k]$ at level k.

$$
\boxed{q_1,\ 2} \xrightarrow{\quad x_2 > 2x_1^2,\ \begin{array}{l} (x_4 := 0) \\ (x_3 := 0) \\ x_2 := x_1^2 - x_1 \\ (x_1 := x_1) \end{array} \quad} \boxed{q_2,\ 4}
$$

Updates for increasing levels $k \leq k'$

Level $i > k$: reset
Level k: unchanged or polynomial update $x_k := P$ for some $P \in \mathbb{Q}[x_1, \ldots, x_{k-1}]$
Level $i < k$: unchanged.

PolITA: Syntax

clocks $X = \{x_1, \dots, x_n\}$ with x_k active at level k,
set of modes Q with $\lambda : Q \to \{1, \dots, n\}$ the state level,
Guards: conjunctions of polynomial constraints
$P \bowtie 0$ with $\bowtie$ in $\{<, \leq, =, \geq, >\}$, and $P \in \mathbb{Q}[x_1, \dots, x_k]$ at level k.

$$
\begin{tikzcd}[column sep=huge]
\boxed{q_1,\ 2} \arrow[r, "{x_2 > 2x_1^2,\ \begin{array}{l}(x_4 := 0)\\ (x_3 := 0)\\ x_2 := x_1^2 - x_1\\ (x_1 := x_1)\end{array}}"] & \boxed{q_2,\ 4} \arrow[r, "{x_4 = 3x_1^2 x_2 + x_3,\ \begin{array}{l}(x_4 := 0)\\ (x_3 := x_3)\\ (x_2 := x_2)\\ (x_1 := x_1)\end{array}}"] & \boxed{q_3,\ 3}
\end{tikzcd}
$$

Updates for decreasing levels $k > k'$

Level $i > k'$: reset
Otherwise: unchanged.

PolITA: semantics

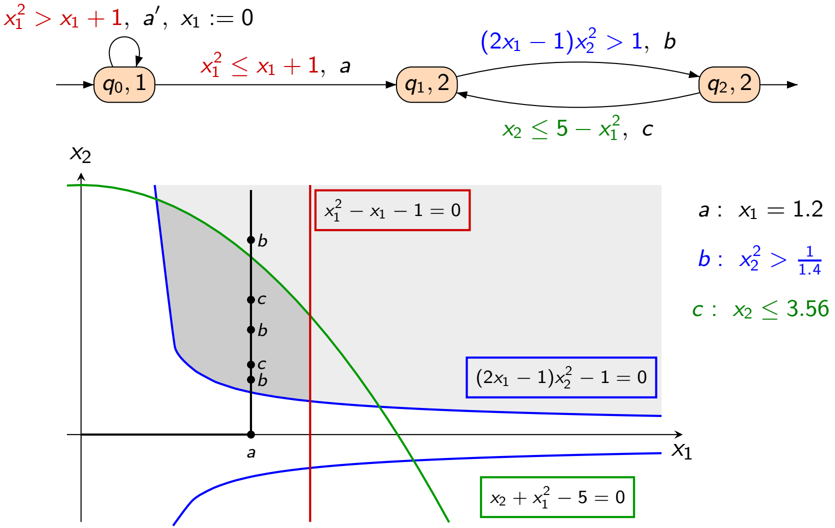
Clock valuations: $v = (v(x_1), \ldots, v(x_n)) \in \mathbb{R}^n$

The semantics of $\mathcal{A}$ is the transition system $\mathcal{T}_\mathcal{A}$

- **configurations** $S = Q \times \mathbb{R}^n$, initial configuration $s_0 = (q_0, \mathbf{0})$
- **time steps** from q at level k: $(q, v) \xrightarrow{d} (q, v +_k d)$, only x_k is active, with all clock values in $v +_k d$ unchanged except $(v +_k d)(x_k) = v(x_k) + d$
- **discrete steps** $(q, v) \xrightarrow{e} (q', v')$ for a transition $e : q \xrightarrow{g,a,u} q'$ if v satisfies the guard g and $v' = v[u]$.

An execution is a sequence alternating time and discrete steps.

Semantics: example



$(q_0, 0, 0) \xrightarrow{1.2} (q_0, 1.2, 0) \xrightarrow{a} (q_1, 1.2, 0) \xrightarrow{0.97} (q_1, 1.2, 0.97) \xrightarrow{b} (q_2, 1.2, 0.97) \ldots$

Blue and green curves meet at real roots of $-2x^5 + x_1^4 + 20x_1^3 - 10x_1^2 - 50x_1 + 26$.

Reachability problem for PolITA

Build a finite automaton $Reg(\mathcal{A})$ time-abstract bisimilar to $\mathcal{T}_\mathcal{A}$

- states: (q, C) for suitable sets of valuations $C \subseteq \mathbb{R}^n$, where polynomials of $\mathcal{A}$ have constant sign (and number of roots),
- abstract time steps: $(q, C) \to (q, succ(C))$ consistent with time elapsing in $\mathcal{T}_\mathcal{A}$,
- discrete steps: $(q, C) \xrightarrow{e} (q', C')$ consistent with discrete transitions in $\mathcal{T}_\mathcal{A}$.

The sets C will be cells from a cylindrical decomposition (CAD) adapted to the polynomials in $\mathcal{A}$.

CAD: basic example

The decomposition starts from a set of polynomials and proceeds in two phases: Elimination phase and Lifting phase.

Starting from single polynomial $P_3 = x_1^2 + x_2^2 + x_3^2 - 1 \in \mathbb{Q}[x_1, x_2][x_3]$

Elimination phase

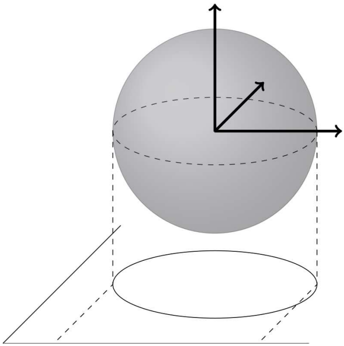
Produces polynomials in $\mathbb{Q}[x_1, x_2]$ and $\mathbb{Q}[x_1]$ required to determine the sign of P_3.

- First polynomial $P_2 = x_1^2 + x_2^2 - 1$ is produced.
  - If $P_2 > 0$ then P_3 has no real root.
  - If $P_2 = 0$ then P_3 has 0 as single root.
  - If $P_2 < 0$ then P_3 has two real roots.
- In turn the sign of $P_2 \in \mathbb{Q}[x_1][x_2]$ depends on $P_1 = x_1^2 - 1$.

Lifting phase

Produces partitions of $\mathbb{R}$, $\mathbb{R}^2$ and $\mathbb{R}^3$ organized in a tree of cells where the signs of these polynomials (in $\{-1, 0, 1\}$) are constant.

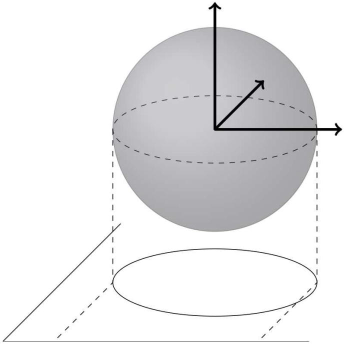
Lifting phase



Level 1 : partition of $\mathbb{R}$ in 5 cells

$C_{-\infty} =]-\infty, -1[$, $C_{-1} = \{-1\}$, $C_0 =]-1, 1[$, $C_1 = \{1\}$, $C_{+\infty} =]1, +\infty[$

Lifting phase



Level 2 : partition of $\mathbb{R}^2$

Above $C_{-\infty}$: a single cell $C_{-\infty} \times \mathbb{R}$

Above C_{-1}: three cells

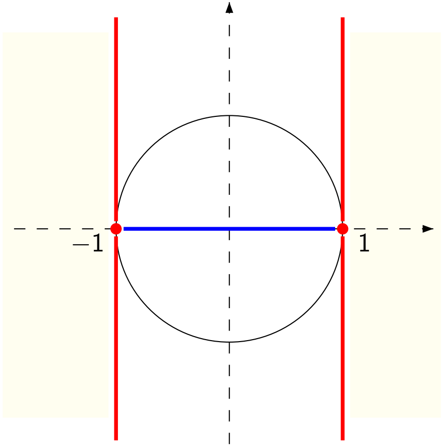
$\{-1\}\times]-\infty, 0[$, $\{(-1, 0)\}$, $\{-1\}\times]0, +\infty[$

Level 1 : partition of $\mathbb{R}$ in 5 cells

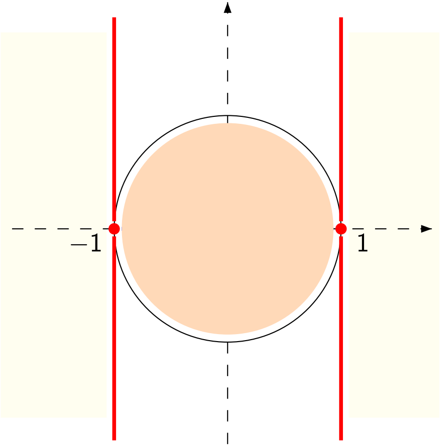
$C_{-\infty} =]-\infty, -1[$, $C_{-1} = \{-1\}$, $C_0 =]-1, 1[$,

$C_1 = \{1\}$, $C_{+\infty} =]1, +\infty[$

Level 2 above C_0

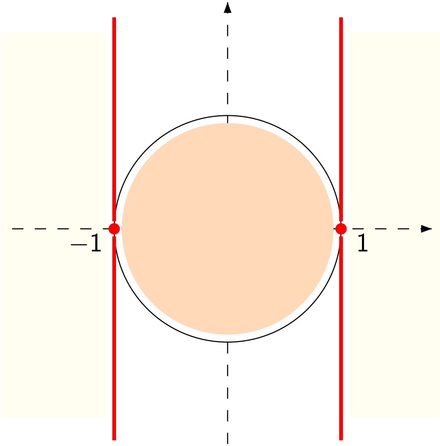


Level 2 above C_0



$\boxed{C_{0,0}}$ $\begin{cases} -1 < x_1 < 1 \\ -\sqrt{1-x_1^2} < x_2 < \sqrt{1-x_1^2} \end{cases}$

Level 2 above C_0

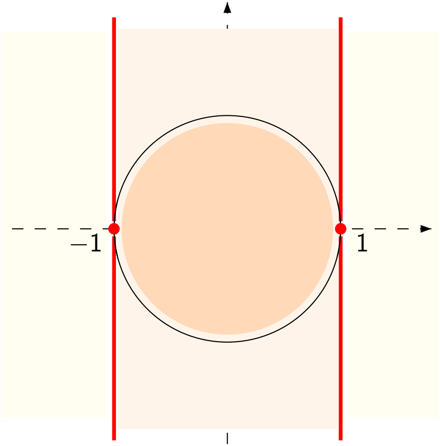


$C_{0,1}$ $\begin{cases} -1 < x_1 < 1 \\ x_2 = \sqrt{1 - x_1^2} \end{cases}$

$\boxed{C_{0,0}}$ $\begin{cases} -1 < x_1 < 1 \\ -\sqrt{1 - x_1^2} < x_2 < \sqrt{1 - x_1^2} \end{cases}$

$C_{0,-1}$ $\begin{cases} -1 < x_1 < 1 \\ x_2 = -\sqrt{1 - x_1^2} \end{cases}$

Level 2 above C_0



$\boxed{C_{0,+\infty}}$ $\begin{cases} -1 < x_1 < 1 \\ x_2 > \sqrt{1 - x_1^2} \end{cases}$

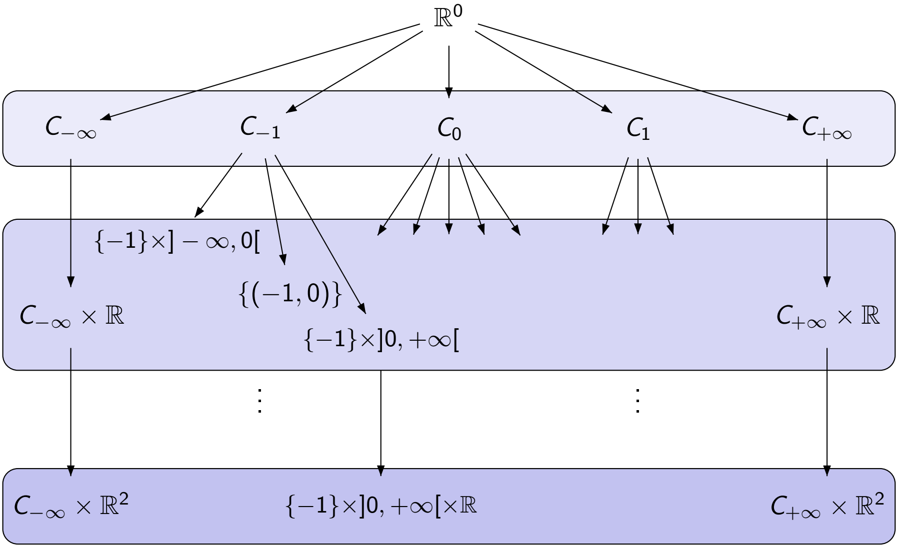
$C_{0,1}$ $\begin{cases} -1 < x_1 < 1 \\ x_2 = \sqrt{1 - x_1^2} \end{cases}$

$\boxed{C_{0,0}}$ $\begin{cases} -1 < x_1 < 1 \\ -\sqrt{1 - x_1^2} < x_2 < \sqrt{1 - x_1^2} \end{cases}$

$C_{0,-1}$ $\begin{cases} -1 < x_1 < 1 \\ x_2 = -\sqrt{1 - x_1^2} \end{cases}$

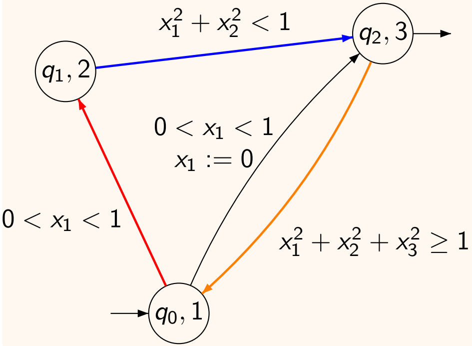
$\boxed{C_{0,-\infty}}$ $\begin{cases} -1 < x_1 < 1 \\ x_2 < -\sqrt{1 - x_1^2} \end{cases}$

The tree of cells



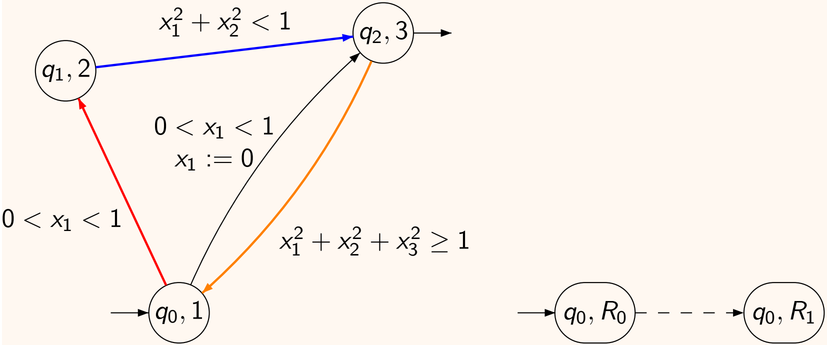
Building the quotient

using the sphere case with some refinements:



Building the quotient

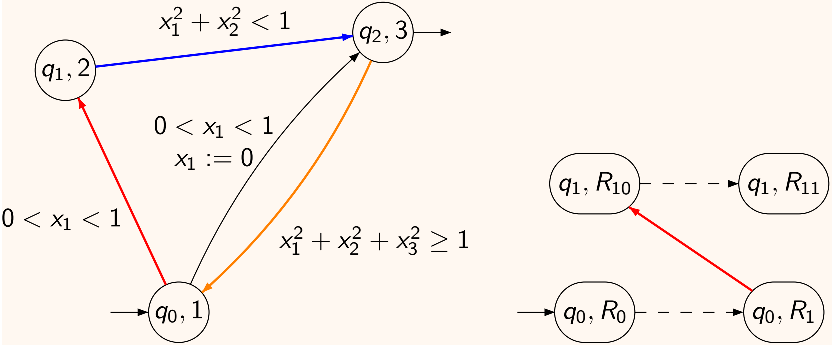
using the sphere case with some refinements:



level 1: $R_0 = (x_1 = 0)$, $R_1 = (0 < x_1 < 1)$,

Building the quotient

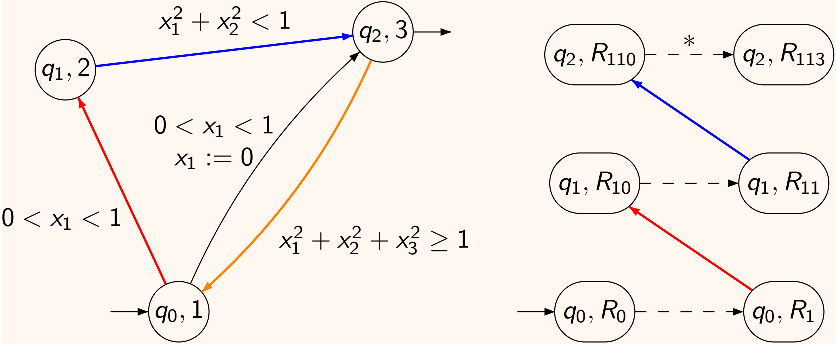
using the sphere case with some refinements:



level 1: $R_0 = (x_1 = 0)$, $R_1 = (0 < x_1 < 1)$,
level 2 above R_1: $R_{10} = (R_1, x_2 = 0)$, $R_{11} = (R_1, 0 < x_2 < \sqrt{1 - x_1^2})$,

Building the quotient

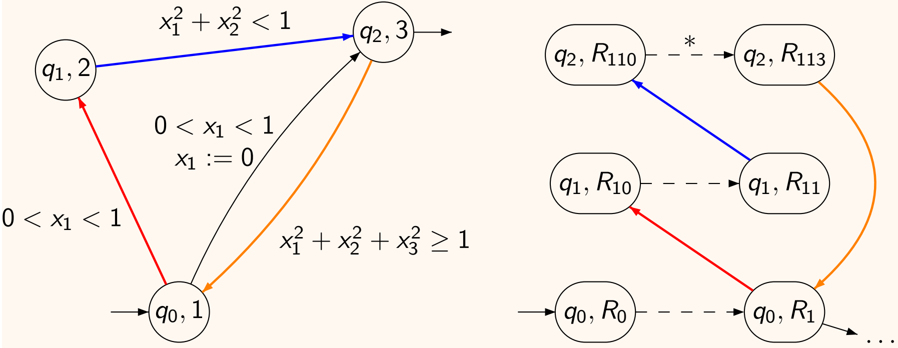
using the sphere case with some refinements:



level 1: $R_0 = (x_1 = 0)$, $R_1 = (0 < x_1 < 1)$,
level 2 above R_1: $R_{10} = (R_1, x_2 = 0)$, $R_{11} = (R_1, 0 < x_2 < \sqrt{1 - x_1^2})$,
level 3 above R_{11}: $R_{110} = (R_{11}, x_3 = 0)$, $R_{111} = (R_{11}, 0 < x_3 < \sqrt{1 - x_1^2 - x_2^2})$,
$R_{112} = (R_{11}, x_3 = \sqrt{1 - x_1^2 - x_2^2})$, $R_{113} = (R_{11}, x_3 > \sqrt{1 - x_1^2 - x_2^2})$,

Building the quotient

using the sphere case with some refinements:



level 1: $R_0 = (x_1 = 0)$, $R_1 = (0 < x_1 < 1)$,
level 2 above R_1: $R_{10} = (R_1, x_2 = 0)$, $R_{11} = (R_1, 0 < x_2 < \sqrt{1 - x_1^2})$,
level 3 above R_{11}: $R_{110} = (R_{11}, x_3 = 0)$, $R_{111} = (R_{11}, 0 < x_3 < \sqrt{1 - x_1^2 - x_2^2})$,
$R_{112} = (R_{11}, x_3 = \sqrt{1 - x_1^2 - x_2^2})$, $R_{113} = (R_{11}, x_3 > \sqrt{1 - x_1^2 - x_2^2})$,
and back to level 1

Effective construction: Elimination

From an initial set of polynomials, the elimination phase produces in 2EXPTIME a family of polynomials $\mathcal{P} = \{\mathcal{P}_k\}_{k \leq n}$ with $\mathcal{P}_k \subseteq \mathbb{Q}[x_1, \ldots, x_k]$ for level k.

Some polynomials do not always have the same degree and roots.
For instance, $B = (2x_1 - 1)x_2^2 - 1$ is of degree 2 in x_2 if and only if $x_1 \neq \frac{1}{2}$.

For $\mathcal{A}_2$

Starting from $\{x_1, A\}$ and $\{x_2, B, C\}$ with $A = x_1^2 - x_1 - 1$ and $C = x_2 + x_1^2 - 5$ results in

- $\mathcal{P}_1 = \{x_1, A, D, E, F, G\}$,
- $\mathcal{P}_2 = \{x_2, B, C\}$,

with $D = 2x_1 - 1$, $E = x_1^2 - 5$, $F = -2x_1^5 + x_1^4 + 20x_1^3 - 10x_1^2 - 50x_1 + 26$, $G = 4(2x_1 - 1)^2$

Effective construction: Lifting

To build the tree of cells in the lifting phase, we need a suitable representation of the roots of these polynomials (and the intervals between them), obtained by iteratively increasing the level.

A description like $x_3 > \sqrt{1 - x_1^2 - x_2^2}$ cannot be obtained in general.

- A point is coded by "the n^{th} root of P".
- The interval $](n, P), (m, Q)[$ is coded by a root of $(PQ)'$.

This lifting phase can be performed on-the-fly, producing only the reachable part of the quotient automaton $Reg(\mathcal{A})$.

Outline

Timed Automata from Alur, Dill (1990)

Polynomial Interrupt Timed Automata

- Reachability using cylindrical decomposition
- Algorithmic issues

A result on Dynamical Systems

Dynamical systems

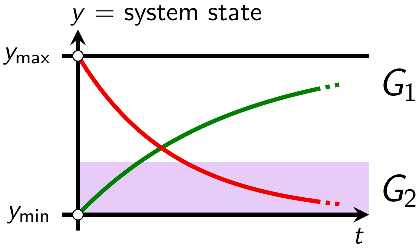
A dynamical system is a hybrid system with:

- a single system mode,
- several possible trajectories,
  hence non-deterministic choice when more than one are available,
- and guards.

Dynamical systems

A dynamical system is a hybrid system with:

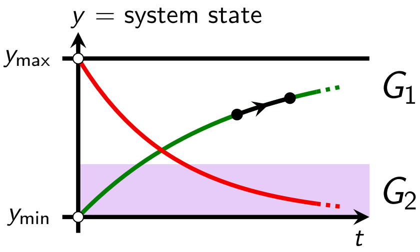
- a single system mode,
- several possible trajectories,
  hence non-deterministic choice when more than one are available,
- and guards.



Dynamical systems

A dynamical system is a hybrid system with:

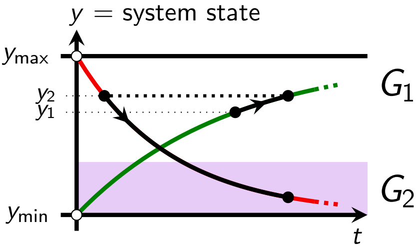
- a single system mode,
- several possible trajectories,
  hence non-deterministic choice when more than one are available,
- and guards.



Dynamical systems

A dynamical system is a hybrid system with:

- a single system mode,
- several possible trajectories,
  hence non-deterministic choice when more than one are available,
- and guards.



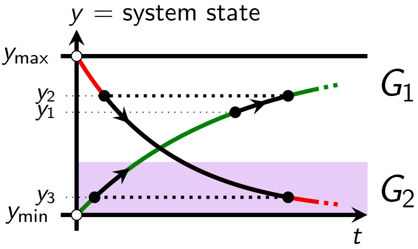
$$y_1 = f(t_1) \rightarrow y_2 = f(t_2) = g(t_3) \rightarrow y_3 = g(t_4)$$

$$t_1 \leq t_2 \qquad\qquad t_3 \leq t_4$$

Dynamical systems

A dynamical system is a hybrid system with:

- a single system mode,
- several possible trajectories,
  hence non-deterministic choice when more than one are available,
- and guards.



Transition system:

$$y_1 = f(t_1) \rightarrow y_2 = f(t_2) = g(t_3) \rightarrow y_3 = g(t_4)$$

$$t_1 \leq t_2 \qquad\qquad t_3 \leq t_4$$

Notations and examples

A dynamical system $(\mathcal{M}, \gamma)$:

- $\mathcal{M} = \langle M, \leqslant, ... \rangle$ a linearly ordered structure,
- $\gamma : V_1 \times V \to V_2$ for $V_1 \subseteq M^{k_1}$, $V \subseteq M$, $V_2 \subseteq M^{k_2}$, all (FO-)definable in $\mathcal{M}$,

and a finite set of guards: definable subsets of V_2.

Notations and examples

A dynamical system $(\mathcal{M}, \gamma)$:

- $\mathcal{M} = \langle M, \leqslant, ... \rangle$ a linearly ordered structure,
- $\gamma : V_1 \times V \to V_2$ for $V_1 \subseteq M^{k_1}$, $V \subseteq M$, $V_2 \subseteq M^{k_2}$, all (FO-)definable in $\mathcal{M}$,

and a finite set of guards: definable subsets of V_2.

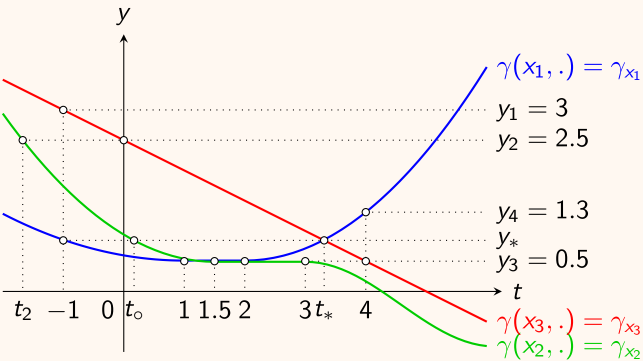
Clocks have dynamics $\gamma : \mathbb{R}_+^n \times [0, +\infty[\to \mathbb{R}_+^n$ with $\gamma(v, t) = \gamma_v(t) = v + t$.

Notations and examples

A dynamical system $(\mathcal{M}, \gamma)$:

- $\mathcal{M} = \langle M, \leqslant, \ldots \rangle$ a linearly ordered structure,
- $\gamma : V_1 \times V \to V_2$ for $V_1 \subseteq M^{k_1}$, $V \subseteq M$, $V_2 \subseteq M^{k_2}$, all (FO-)definable in $\mathcal{M}$,

and a finite set of guards: definable subsets of V_2.



Bisimulations for dynamical systems

Bisimulations:

- Splitting system states (V_2) according to similar behaviours (consistent with guards and time elapsing)
- k-step bisimulation: similar behaviours up to k steps.

Bisimulations for dynamical systems

Bisimulations:

- Splitting system states (V_2) according to similar behaviours (consistent with guards and time elapsing)
- k-step bisimulation: similar behaviours up to k steps.

Bisimulation is undecidable

but under mild assumptions, k-step bisimulation is decidable for all $k \geq 0$.

Bisimulations for dynamical systems

Bisimulations:

- Splitting system states (V_2) according to similar behaviours (consistent with guards and time elapsing)
- k-step bisimulation: similar behaviours up to k steps.

Bisimulation is undecidable

but under mild assumptions, k-step bisimulation is decidable for all $k \geq 0$.

Theorem [Lafferriere, Pappas, Sastry 2000]

Bisimulation is decidable and induces a finite partition when:
$\gamma : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is solution of $\mathrm{d}\gamma(x,t)/\mathrm{d}t = F(\gamma(x,t))$ definable in an o-minimal theory of $\mathbb{R}$.

O-minimal structures

A linearly ordered structure $\langle M, \leqslant, \ldots \rangle$ is o-minimal

if every definable set is a finite union of intervals with bounds in $M_{\pm\infty}$.

O-minimal structures

A linearly ordered structure $\langle M, \leqslant, \dots \rangle$ is o-minimal

if every definable set is a finite union of intervals with bounds in $M_{\pm\infty}$.

A few examples: $(\mathbb{R}, \leqslant, +, \times)$, $(\mathbb{Q}, \leqslant, 1, +)$, $(\mathbb{Z}_{\geqslant 0}, \leqslant)$, $(\mathbb{R}, \leqslant, +, \times, \exp)$

O-minimal structures

A linearly ordered structure $\langle M, \leqslant, \dots \rangle$ is o-minimal

if every definable set is a finite union of intervals with bounds in $M_{\pm\infty}$.

A few examples: $(\mathbb{R}, \leqslant, +, \times)$, $(\mathbb{Q}, \leqslant, 1, +)$, $(\mathbb{Z}_{\geqslant 0}, \leqslant)$, $(\mathbb{R}, \leqslant, +, \times, \exp)$

. . . and counter-examples:

- $(\mathbb{Q}, \leqslant, +, \times)$ — $x^2 \leqslant 1 + 1 \Leftrightarrow -\sqrt{2} \leqslant x \leqslant \sqrt{2}$
- $(\mathbb{Z}_{\geqslant 0}, \leqslant, +)$ — $\exists z, x = z + z \Leftrightarrow x$ is even
- $(\mathbb{R}, \leqslant, \sin)$ — $\sin(x) = 0 \Leftrightarrow x \in \pi\mathbb{Z}$

Properties

[Pillay, Steinhorn 88]

Property 1

Let $(M, \leqslant, \ldots)$ be o-minimal and $f : M \to M$ be definable. There exists a finite partition $(\mathcal{I}_1, \ldots, \mathcal{I}_k)$ of M into intervals s.t., for all $j \leqslant k$:

1. $f_{|\mathcal{I}_j}$ is constant, or
2. $f_{|\mathcal{I}_j}$ is one-to-one and monotonic, and $f(\mathcal{I}_j)$ is an interval.

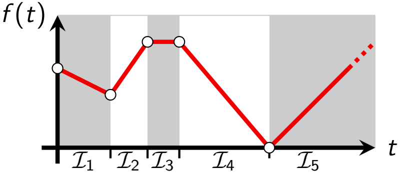
Properties

[Pillay, Steinhorn 88]

Property 1

Let $(M, \leqslant, \ldots)$ be o-minimal and $f : M \to M$ be definable. There exists a finite partition $(\mathcal{I}_1, \ldots, \mathcal{I}_k)$ of M into intervals s.t., for all $j \leqslant k$:

1. $f_{|\mathcal{I}_j}$ is constant, or
2. $f_{|\mathcal{I}_j}$ is one-to-one and monotonic, and $f(\mathcal{I}_j)$ is an interval.



Properties

[Pillay, Steinhorn 88]

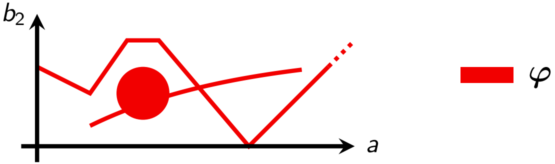
Property 1

Let $(M, \leqslant, \ldots)$ be o-minimal and $f : M \to M$ be definable. There exists a finite partition $(\mathcal{I}_1, \ldots, \mathcal{I}_k)$ of M into intervals s.t., for all $j \leqslant k$:

1. $f_{|\mathcal{I}_j}$ is constant, or
2. $f_{|\mathcal{I}_j}$ is one-to-one and monotonic, and $f(\mathcal{I}_j)$ is an interval.

Property 2

Let φ be an ℓ-variable formula. There exists $\mathbf{N}_\varphi$ s.t., for all $b_2, \ldots, b_\ell \in M$, the set $\{a \in M \mid (a, b_2, \ldots, b_\ell) \models \varphi\}$ is a union of at most $\mathbf{N}_\varphi$ intervals.



Properties

[Pillay, Steinhorn 88]

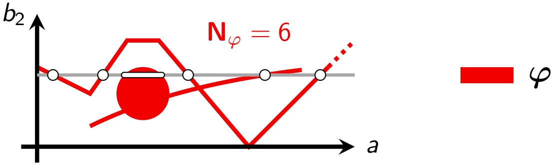
Property 1

Let $(M, \leqslant, \ldots)$ be o-minimal and $f : M \to M$ be definable. There exists a finite partition $(\mathcal{I}_1, \ldots, \mathcal{I}_k)$ of M into intervals s.t., for all $j \leqslant k$:

1. $f_{|\mathcal{I}_j}$ is constant, or
2. $f_{|\mathcal{I}_j}$ is one-to-one and monotonic, and $f(\mathcal{I}_j)$ is an interval.

Property 2

Let φ be an ℓ-variable formula. There exists $\mathbf{N}_\varphi$ s.t., for all $b_2, \ldots, b_\ell \in M$, the set $\{a \in M \mid (a, b_2, \ldots, b_\ell) \models \varphi\}$ is a union of at most $\mathbf{N}_\varphi$ intervals.



Result

[BBJ 18]

Generalising Lafferriere et al.:

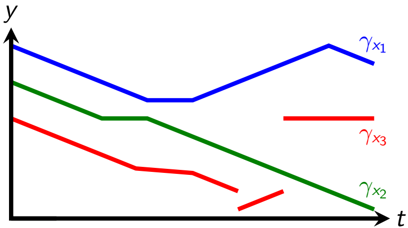
- o-minimal **real** theory $\rightarrow$ **any** o-minimal theory
- trajectories partition $\mathbb{R}^n$ $\rightarrow$ trajectories **may overlap**

Result

[BBJ 18]

Generalising Lafferriere et al.:

- o-minimal **real** theory → **any** o-minimal theory
- trajectories partition $\mathbb{R}^n$ → trajectories **may overlap**

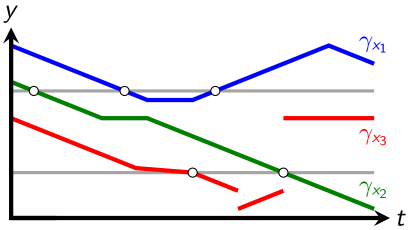


Result

[BBJ 18]

Generalising Lafferriere et al.:

- o-minimal **real** theory $\rightarrow$ **any** o-minimal theory
- trajectories partition $\mathbb{R}^n$ $\rightarrow$ trajectories **may overlap**



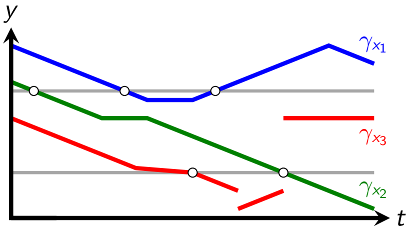
$$\gamma_{x_1}(M) \cap \gamma_{x_2}(M) \neq \emptyset$$
$$\gamma_{x_1}(M) \cap \gamma_{x_3}(M) = \emptyset$$
$$\gamma_{x_2}(M) \cap \gamma_{x_3}(M) \neq \emptyset$$

Result

[BBJ 18]

Generalising Lafferriere et al.:

- o-minimal **real** theory $\rightarrow$ **any** o-minimal theory
- trajectories partition $\mathbb{R}^n$ $\rightarrow$ trajectories **may overlap**



$x_1 \sim x_2$

$x_1 \not\sim x_3$

$x_2 \sim x_3$

$x_1 \sim^* x_3$

Result

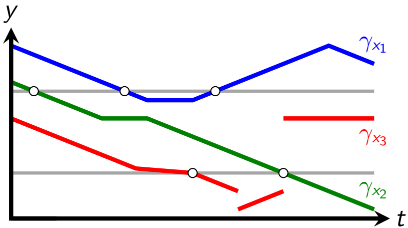
[BBJ 18]

Generalising Lafferriere et al.:

- o-minimal **real** theory $\rightarrow$ **any** o-minimal theory
- trajectories partition $\mathbb{R}^n$ $\rightarrow$ trajectories **may overlap**

In an o-minimal dynamical system

- if $V_1^*(x) \stackrel{\text{def}}{=\!=} \{x' \mid x \sim^* x'\}$ is **finite** for all x, (Finite Crossing)
  the bisimulation relation is **decidable**; (if the theory is decidable)



$x_1 \sim x_2$

$x_1 \not\sim x_3$

$x_2 \sim x_3$

$x_1 \sim^* x_3$

Result

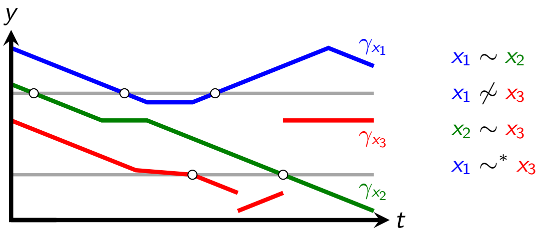
[BBJ 18]

Generalising Lafferriere et al.:

- o-minimal **real** theory $\rightarrow$ **any** o-minimal theory
- trajectories partition $\mathbb{R}^n$ $\rightarrow$ trajectories **may overlap**

In an o-minimal dynamical system

- if $V_1^*(x) \stackrel{\text{def}}{=\!=} \{x' \mid x \sim^* x'\}$ is **finite** for all x, (Finite Crossing)
  the bisimulation relation is **decidable**; (if the theory is decidable)
- if the sizes $|V_1^*(x)|$ are **uniformly bounded**, (Uniform Crossing)
  the bisimulation relation is **definable** and induces **finite** partition.



Idea of the proof

First step: decomposition

For all $x \in V_1$ with dynamics γ_x:

- Produce a classification of time intervals into x-static or x-adaptable intervals.
- If $V_1(x) = \{x' \in V_1 \mid x \sim x'\}$ is finite, then there is a finite definable partition of the time set v into maximal x-static and x-adaptable intervals.
- For those $\mathcal{I}$, all states in $\gamma_x(\mathcal{I})$ are bisimilar.

Idea of the proof

First step: decomposition

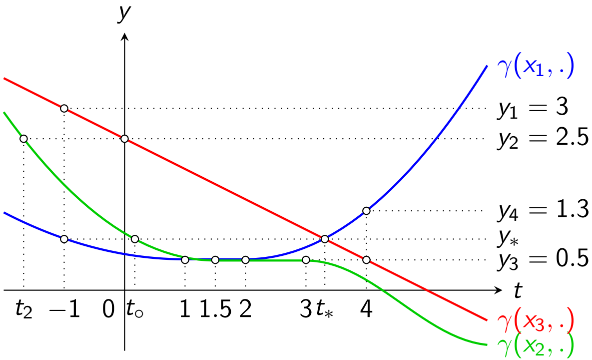
For all $x \in V_1$ with dynamics γ_x:

- Produce a classification of time intervals into x-static or x-adaptable intervals.
- If $V_1(x) = \{x' \in V_1 \mid x \sim x'\}$ is finite, then there is a finite definable partition of the time set v into maximal x-static and x-adaptable intervals.
- For those $\mathcal{I}$, all states in $\gamma_x(\mathcal{I})$ are bisimilar.

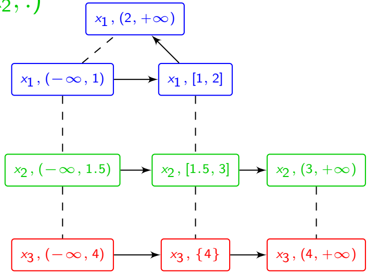
Second step: building a bisimulation graph

- with nodes $(x, \mathcal{I})$ for the intervals above,
- edges $(x, \mathcal{I}) \to (x, \mathcal{J})$ that represent time elapsing on γ_x,
- ε-edges $(x, \mathcal{I}) \to (x', \mathcal{I}')$ that represent jumps between trajectories.

Example



y
t
$\gamma(x_1, .)$
$y_1 = 3$
$y_2 = 2.5$
$y_4 = 1.3$
y_*
$y_3 = 0.5$
t_2 -1 0 $t_\circ$ 1 1.5 2 3 t_* 4
$\gamma(x_3, .)$
$\gamma(x_2, .)$



$x_1, (2, +\infty)$
$x_1, (-\infty, 1)$
$x_1, [1, 2]$
$x_2, (-\infty, 1.5)$
$x_2, [1.5, 3]$
$x_2, (3, +\infty)$
$x_3, (-\infty, 4)$
$x_3, \{4\}$
$x_3, (4, +\infty)$

Conclusion

Summary

- Reachability is decidable in two models without strong resets: Timed Automata and Polynomial Interrupt Timed Automata.
- Bisimulation is decidable in a richer model of dynamical systems, which can immediately be extended with modes and strong resets.

Going further

- Refine the crossing conditions,
- Add modes with weaker jump conditions.

Conclusion

Summary

- Reachability is decidable in two models without strong resets: Timed Automata and Polynomial Interrupt Timed Automata.
- Bisimulation is decidable in a richer model of dynamical systems, which can immediately be extended with modes and strong resets.

Going further

- Refine the crossing conditions,
- Add modes with weaker jump conditions.

Thank you