Verification of Polynomial Interrupt Timed Automata

Béatrice Bérard[1], Serge Haddad[2], Claudine Picaronny[2],
Mohab Safey El Din[1], Mathieu Sassolas[3]

[1]Université P. & M. Curie, LIP6
[2]ENS Cachan, LSV
[3]Université Paris-Est, LACL
[4]CNRS, INRIA

GT ALGA, April 11th, 2016

Context: Verification of hybrid systems

Hybrid automata

Hybrid automaton = finite automaton + variables
Variables evolve in states and can be tested and updated on transitions.

- Clocks are variables with slope 1 in all states
- Stopwatches are variables with slope 0 or 1

Timed automaton = finite automaton + clocks with guards $x \bowtie c$ and reset [Alur, Dill 1990]

Verification problems are mostly undecidable

- Decidability requires restricting either the flows [Henzinger et al. 1998] or the jumps [Alur et al. 2000] for flows $\dot{x} = Ax$
- Other approaches exist like bounded delay reachability or approximations by discrete transition systems.

The model of PolITA

In Polynomial Interrupt Timed Automata (PolITA)

- variables are interrupt clocks, a restricted form of stopwatches, ordered along hierarchical levels,
- guards are polynomial constraints and variables can be updated by polynomials.

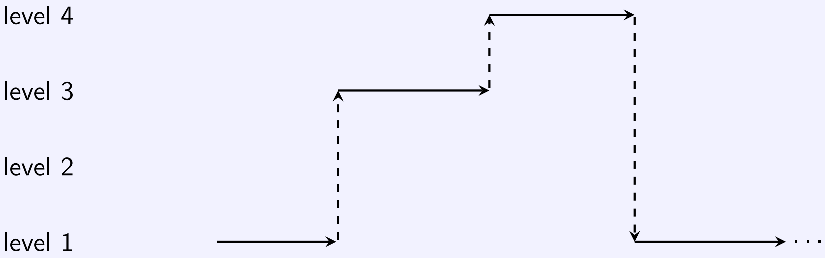
Results

- Reachability is decidable in 2EXPTIME.
- The result still holds for several extensions.
- A restricted form of quantitative model checking is also decidable.
- The class PolITA is incomparable with the class SWA of Stopwatch Automata.

Interrupt clocks

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

Several levels with exactly one active clock at each level

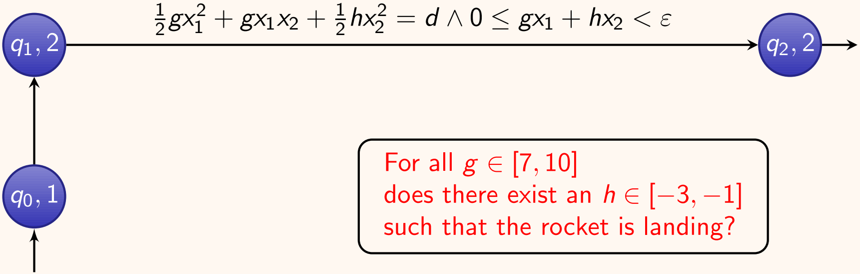


$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad \text{Exec:} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{1.5} \begin{bmatrix} 1.5 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{2.1} \begin{bmatrix} 1.5 \\ 0 \\ 2.1 \\ 0 \end{bmatrix} \xrightarrow{1.7} \begin{bmatrix} 1.5 \\ 0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

Polynomial constraints

Landing a rocket

- First stage (lasting x_1): from distance d, the rocket approaches the land under gravitation g;
- Second stage (lasting x_2): 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?

Polynomial constraints are also used in the modeling of discrete systems.

Outline

Polynomial Interrupt Timed Automata

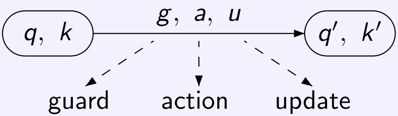
Reachability using cylindrical decomposition

Algorithmic issues

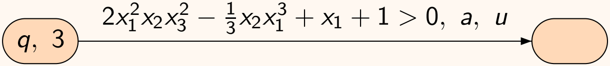
PolITA: syntax

$\mathcal{A} = (\Sigma, Q, q_0, X, \lambda, \Delta)$

- Alphabet Σ, finite set of states Q, initial state q_0,
- set of clocks $X = \{x_1, \ldots, x_n\}$, with x_k for level k,
- $\lambda : Q \to \{1, \ldots, n\}$ state level, with $x_{\lambda(q)}$ the active clock in state q,
- Transitions in Δ:



- Guards: conjunctions of polynomial constraints in $\mathbb{Q}[x_1, \ldots, x_n]$
  $P \bowtie 0$ with $\bowtie$ in $\{<, \leq, =, \geq, >\}$, and $P \in \mathbb{Q}[x_1, \ldots, x_k]$ at level k.



PolITA: updates

From level k to k'

increasing level $k \leq k'$

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.

$$
\begin{tikzcd}
(q_1,\ 2) \arrow[rrr, "{x_2 > 2x_1^2,\ \begin{array}{l}(x_1 := x_1)\\ x_2 := x_1^2 - x_1\\ (x_3 := 0)\\ (x_4 := 0)\end{array}}"] & & & (q_2,\ 4)
\end{tikzcd}
$$

PolITA: updates

From level k to k'

increasing level $k \leq k'$

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.

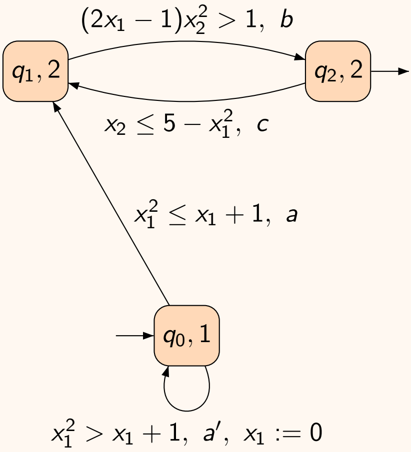
$$
\begin{tikzcd}[column sep=huge]
(q_1,\ 2) \arrow[r, "{x_2 > 2x_1^2,\ \begin{matrix}(x_1 := x_1)\\ x_2 := x_1^2 - x_1\\ (x_3 := 0)\\ (x_4 := 0)\end{matrix}}"] & (q_2,\ 4) \arrow[r, "{x_4 = 3x_1^2 x_2 + x_3,\ \begin{matrix}(x_1 := x_1)\\ (x_2 := x_2)\\ (x_3 := x_3)\\ (x_4 := 0)\end{matrix}}"] & (q_3,\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

Decreasing level

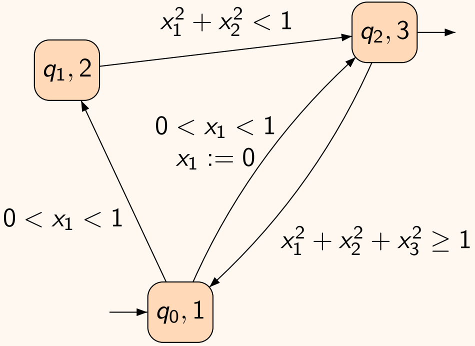
Level $i > k'$: reset
Otherwise: unchanged.

Examples

$\mathcal{A}_2$ in dimension 2



$\mathcal{A}_3$ in dimension 3



PolITA: semantics

Clock valuation

$v = (v(x_1), \dots, v(x_n)) \in \mathbb{R}^n$

A transition system $\mathcal{T}_\mathcal{A} = (S, s_0, \rightarrow)$ for $\mathcal{A} = (\Sigma, Q, q_0, X, \lambda, \Delta)$

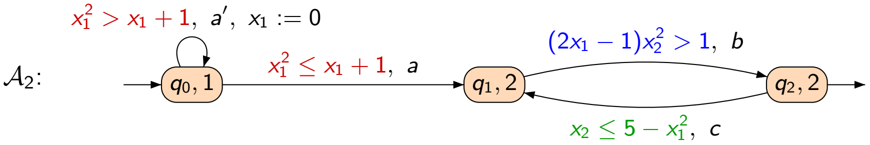
- **configurations** $S = Q \times \mathbb{R}^n$, initial configuration $s_0 = (q_0, v_0)$ with $v_0 = \mathbf{0}$
- **time steps** from q at level k: $(q, v) \xrightarrow{d} (q, v +_k d)$, only x_k is active, with all clock values in $v +_k d$ unchanged except $(v +_k d)(x_k) = v(x_k) + d$
- **discrete steps** $(q, v) \xrightarrow{e} (q', v')$ for a transition $e : q \xrightarrow{g,a,u} q'$ if v satisfies the guard g and $v' = v[u]$.

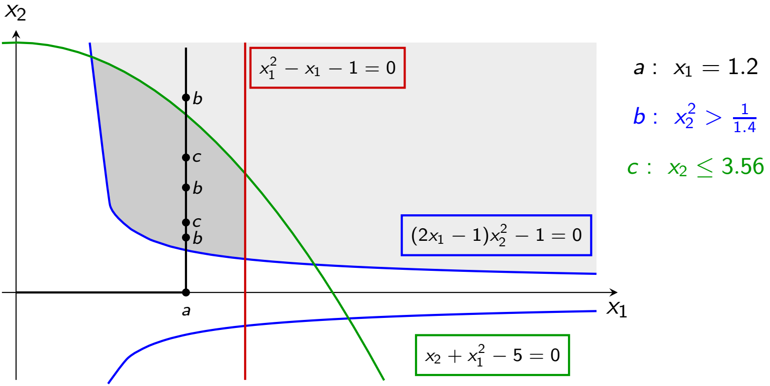
An execution

alternates time and discrete steps

$(q_0, v_0) \xrightarrow{d_0} (q_0, v_0 +_{\lambda(q_0)} d_0) \xrightarrow{e_0} (q_1, v_1) \xrightarrow{d_1} (q_1, v_1 +_{\lambda(q_1)} d_1) \xrightarrow{e_1} \dots$

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

Given $\mathcal{A} = (\Sigma, Q, q_0, X, \lambda, \Delta)$ 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 ?

Build a finite quotient automaton $\mathcal{R}_\mathcal{A}$

time-abstract bisimilar to $\mathcal{T}_\mathcal{A}$:

- states of $\mathcal{R}_\mathcal{A}$ are of the form (q, C) for suitable sets of valuations $C \subseteq \mathbb{R}^n$, where polynomials of $\mathcal{A}$ have constant sign (and number of roots),
- time abstract transitions of $\mathcal{R}_\mathcal{A}$: $(q, C) \to (q, succ(C))$ where $succ(C)$ is the time successor of C, consistent with time elapsing in $\mathcal{T}_\mathcal{A}$,
- discrete transitions of $\mathcal{R}_\mathcal{A}$: $(q, C) \xrightarrow{e} (q', C')$ for $e : q \xrightarrow{g,a,u} q'$ in Δ if C satisfies the guard g and $C' = C[u]$, consistent with discrete steps in $\mathcal{T}_\mathcal{A}$.

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

Cylindrical decomposition: 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

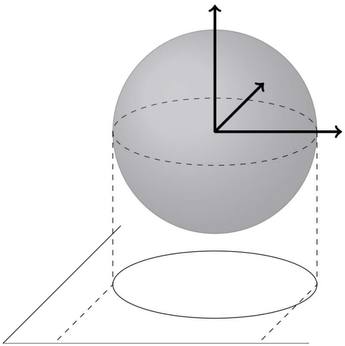
Produces polynomials in $\mathbb{Q}[x_1, x_2]$ and $\mathbb{Q}[x_1]$ required to determine the sign of P_3.

- First polynmial $P_2 = x_1^2 + x_2^2 - 1$ 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.

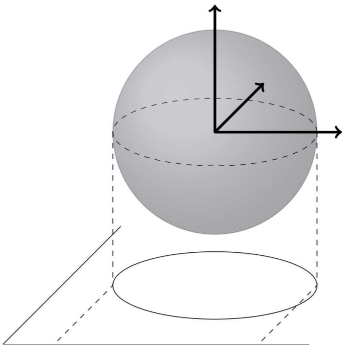
Lifting phase



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

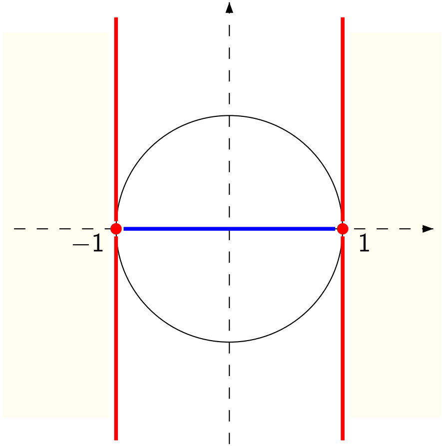
$\{-1\}\times]-\infty, 0[$, $\{(-1, 0)\}$, $\{-1\}\times]0, +\infty[$

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

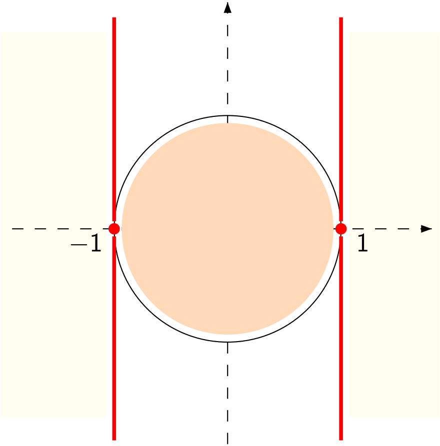
$C_{-\infty} =]-\infty, -1[$, $C_{-1} = \{-1\}$, $C_0 =]-1, 1[$,

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

Level 2 above C_0

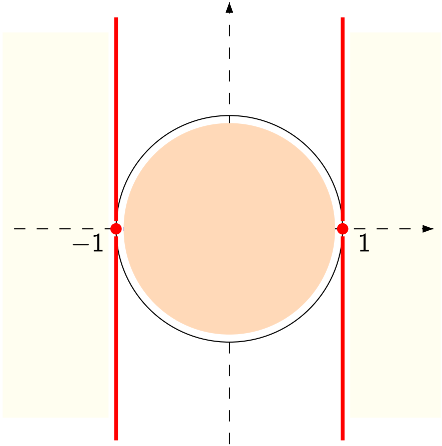


Level 2 above C_0



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

Level 2 above C_0

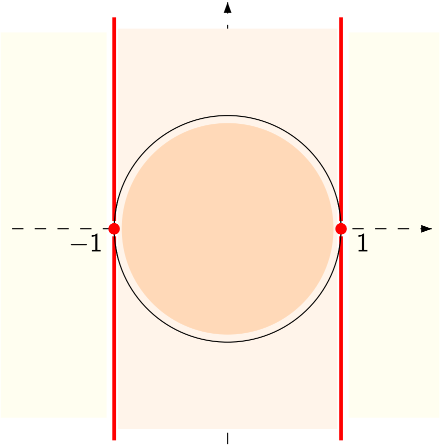


$C_{0,1}$ $\begin{cases} -1 < x_1 < 1 \\ x_2 = \sqrt{1 - x_1^2} \end{cases}$

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

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

Level 2 above C_0



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

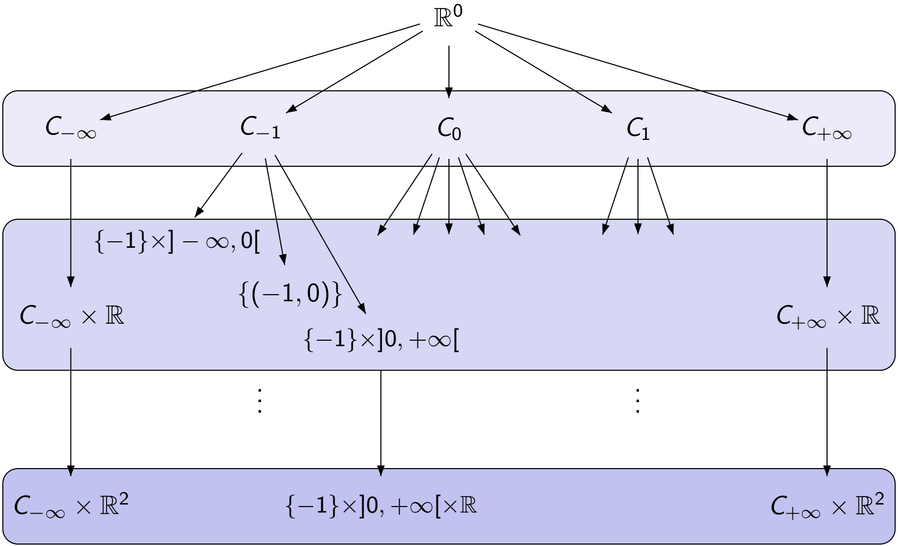
$C_{0,1}$ $\begin{cases} -1 < x_1 < 1 \\ x_2 = \sqrt{1 - x_1^2} \end{cases}$

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

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

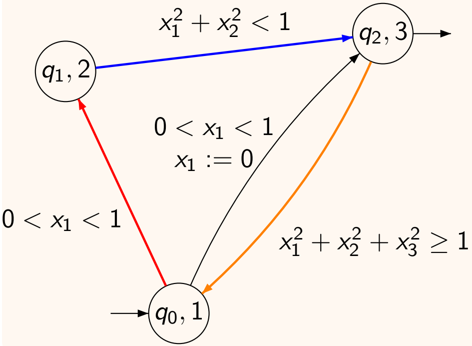
$\boxed{C_{0,-\infty}}$ $\begin{cases} -1 < x_1 < 1 \\ x_2 < -\sqrt{1 - x_1^2} \end{cases}$

The tree of cells



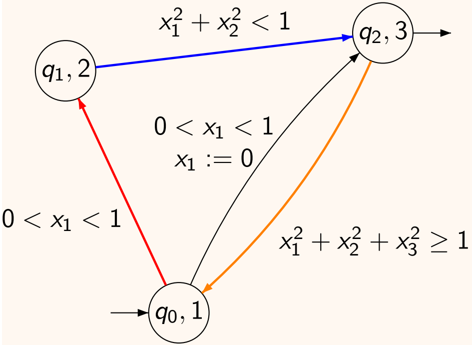
Building the quotient

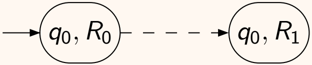
partially, for $\mathcal{A}_3$, using the sphere case with some refinements:



Building the quotient

partially, for $\mathcal{A}_3$, using the sphere case with some refinements:

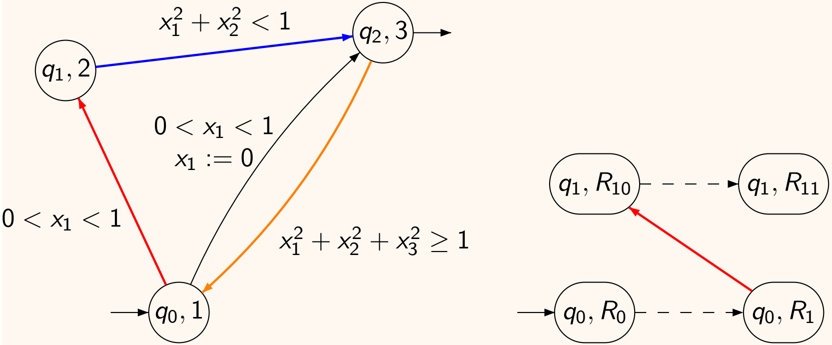




level 1: $R_0 = (x_1 = 0)$, $R_1 = (0 < x_1 < 1)$,

Building the quotient

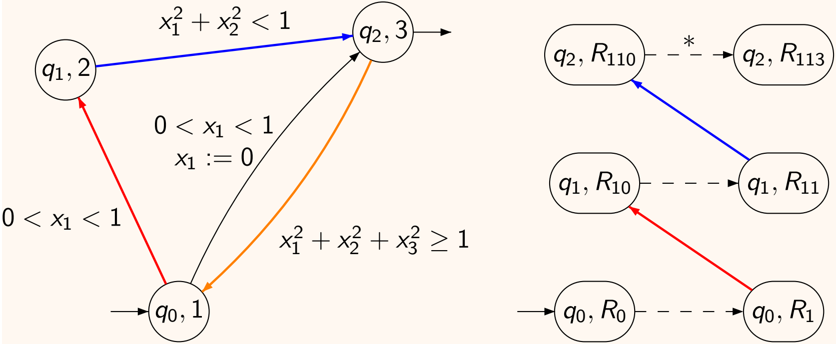
partially, for $\mathcal{A}_3$, using the sphere case with some refinements:



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

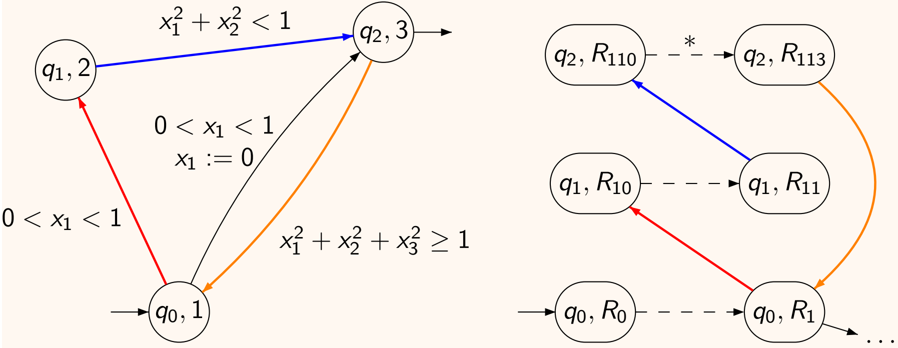
partially, for $\mathcal{A}_3$, using the sphere case with some refinements:



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

partially, for $\mathcal{A}_3$, 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 have always 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 $\mathcal{R}_{\mathcal{A}}$.

Conclusion

In the class PolITA

- Reachability is decidable in 2EXPTIME.
- The untimed language of a PolITA (with final states) is regular.
- Model checking is decidable for a quantitative version of CTL using polynomial constraints on the automaton clocks.
- Guards can be extended by adding parameters, auxiliary clocks, and updates can be done at levels lower than the current level.
- PolITA and Stopwatch Automata are incomparable w.r.t. timed language acceptance.

Future work

- Experiments, thanks to Rémi Garnier and Mathieu Huot (L3 students of ENS Cachan) who developped a prototype.
- Adapt more efficient methods for quantifier elimination.
- Extension to o-minimal decidable theories.

Thank you