Interrupt Timed Automata and their Extensions

B. Bérard, S. Haddad, A. Jovanović, D. Lime, C. Picaronny,
M. Safey El Din, M. Sassolas

ANR MAVERIQ, March 9th, 2022
(based on previous talks given by some of the authors)

Interrupt Timed Automata **(FOSSACS'09, TIME'10, FMSD'12)**

Parametric Interrupt Timed Automata **(RP'13, FI'16)**

Polynomial Interrupt Timed Automata **(RP'15, IC'21, IPL'21)**

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.

Outline

Interrupt Timed Automata (FOSSACS'09, TIME'10, FMSD'12)

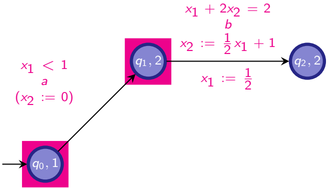
Parametric Interrupt Timed Automata (RP'13, FI'16)

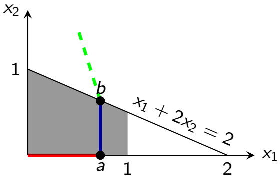
Polynomial Interrupt Timed Automata (RP'15, IC'21, IPL'21)

Interrupt Timed Automata (ITA)

Each state q has an integer *level* $\lambda(q)$. There is one clock x_k per level k.

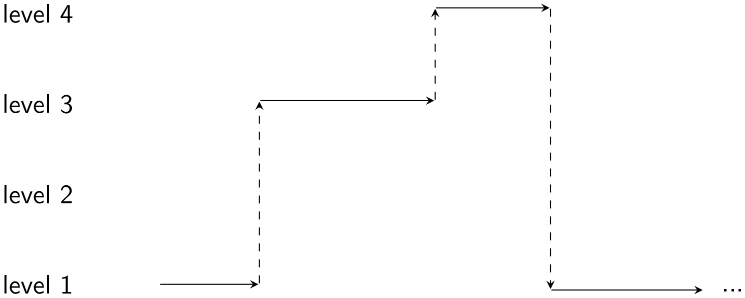
- At a given level, the clock associated with it is active (rate 1)
  - clocks of lower levels are suspended (rate 0)
  - clocks of higher levels are not yet activated





- Guards are affine constraints on the clocks of levels lower than or equal of the current level
- A transition can update the values of clocks
  - level ↑: clocks relevant before can be left unchanged or take an affine expression of clocks of strictly lower level, clocks relevant after are reset;
  - level ↓: clocks relevant after can be left unchanged or take an affine expression of clocks of strictly lower level.

Behaviour of an ITA



$$\begin{bmatrix} x_1 = 0 \\ x_2 = 0 \\ x_3 = 0 \\ x_4 = 0 \end{bmatrix} \xrightarrow{1.5} \begin{bmatrix} 1.5 \\ 00 \\ 00 \\ 0 \end{bmatrix} \xrightarrow{2.1} \begin{bmatrix} 1.5 \\ 0 \\ 2.1 \\ 00 \end{bmatrix} \xrightarrow{1.7} \begin{bmatrix} 1.5 \\ 0 \\ 2.1 \\ 1.7 \end{bmatrix} \xrightarrow{\varepsilon} \begin{bmatrix} 1.5 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{2.2} \begin{bmatrix} 3.7 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Regularity of the untimed language

First step. Construction of a family $(E_k)_{k \leq n}$ where E_k is a set of affine expressions including 0, x_k, and expressions $\sum_{i<k} a_i x_i + b$ by a saturation process that:

- takes into account the guards $x_k \bowtie \sum_{i<k} a_i x_i + b$ of transitions;
- generates new expressions by applying the updates of transitions;
- generates new expressions by considering differences of expressions at higher levels.

Second step. Construction of an automaton

- whose states are pairs $(q, (\sim_k)_{k \leq \lambda(q)})$ where $\sim_k$ is a total preorder over E_k;
- whose transitions are either discrete or timed transitions which can be effectively built due the saturated feature of $(E_k)_{k \leq n}$.

This automaton accepts the untimed language of the ITA.

Some timed temporal logics

- $\mathsf{TCTL}_c^{\mathrm{int}}$ is defined by the following grammar:

$$\psi ::= p \mid \psi \wedge \psi \mid \neg\psi \mid \sum_{i \geq 1} a_i \cdot x_i + b \bowtie 0 \mid \mathbf{A}\ \psi \mathbf{U}\ \psi \mid \mathbf{E}\ \psi \mathbf{U}\ \psi$$

where $p \in AP$ is an atomic proposition, x_i are model clocks, a_i and b are rational numbers and $\bowtie \in \{>, \geq, =, \leq, <\}$.

Example. $\mathbf{A}\ (x_2 \leq 3)\ \mathbf{U}\ \mathit{safe}$ expresses that all executions reach a safe state while spending less than 3 time units in level 2.

(assuming x_2 is not updated during the execution)

- TCTL_p is defined by the following grammar:

$\varphi_p := p \mid \varphi_p \wedge \varphi_p \mid \neg\varphi_p$ and $\psi := \psi \wedge \psi \mid \neg\psi \mid \varphi_p \mid \mathbf{A}\ \varphi_p\ \mathbf{U}_{\bowtie a}\ \varphi_p \mid \mathbf{E}\ \varphi_p\ \mathbf{U}_{\bowtie a}\ \varphi_p$

where $p \in AP$ is an atomic proposition, $a \in \mathbb{Q}^+$, and $\bowtie \in \{>, \geq, \leq, <\}$.

Examples.

The system is error free for at least 50 t.u. is expressed by $\mathbf{A}\ (\neg \mathit{error})\ \mathbf{U}_{\geq 50}\ \top$.

The system will reach a safe state within 7 t.u. is expressed by $\mathbf{A}\ \mathbf{F}_{\leq 7}\ \mathit{safe}$.

Complexity of model checking

The building of the automaton can be performed:

- in 2-EXPTIME;
- in PTIME when the number of clocks is fixed.

Model-checking is achieved with the same complexity by:

- adapting the automaton construction ;
- and adding information relevant to the formula ;
- then performing CTL model checking on the automaton.

The reachability problem

In an ITA$^-$, only the clock of the current level may be updated.

An ITA can be simulated by an ITA$^-$ with the same clocks such that its number of edges/states is exponential w.r.t. the number of edges/states of the ITA.

In an ITA$^-$, a state is reachable if and only if it is reachable in an number of steps exponential w.r.t. the number of clocks and polynomial w.r.t. the number of edges and states.

A non deterministic reachability decision procedure.

- Convert the ITA into an ITA$^-$;
- Guess a sequence of transitions;
- Solve a linear programming problem.

The reachability problem belongs to NEXPTIME.

Other results

The families of timed languages of ITA and TA are incomparable.

The model checking problem of State Clock Logic (SCL) is undecidable.

Conjecture. The model checking problem of TCTL is undecidable.

ITA and TA can be combined with decidable properties including reachability.

Outline

Interrupt Timed Automata (FOSSACS'09, TIME'10, FMSD'12)

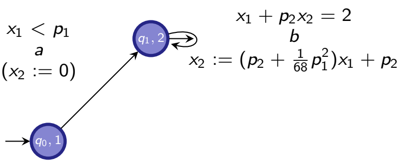
Parametric Interrupt Timed Automata (RP'13, FI'16)

Polynomial Interrupt Timed Automata (RP'15, IC'21, IPL'21)

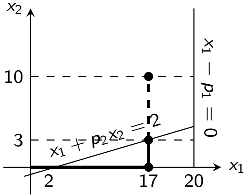
Parametric ITA (PITA)

Polynomial parametric expressions $C = \Sigma_{x \in X} a_x x + b$:

- additive parametrization: $x_1 - p_1 < 0$
- multiplicative parametrization: $x_1 + p_2 x_2 - 2 = 0$



(a) A PITA $\mathcal{A}$ with two interrupt levels



(b) A possible trajectory in $\mathcal{A}$

- $(q_0, 0, 0) \xrightarrow{17} (q_0, 17, 0) \xrightarrow{a} (q_1, 17, 0) \xrightarrow{3} (q_1, 17, 3) \xrightarrow{b} (q_1, 17, 18p_2 + \frac{17}{68} p_1^2)$
- parameter valuation $\pi : p_1 = -5, p_2 = 20$

Analysis of PITA

Existential Reachability Problem

Does there exist a parameter valuation such that some q is reachable from q_0 for a given PITA $\mathcal{A}$?

Universal Reachability Problem

Is q is reachable from q_0, in a given PITA $\mathcal{A}$, for all parameter valuations?

Robust Reachability Problem

Does there exist a parameter valuation π and a real $\varepsilon > 0$ such that for all π', with $||\pi - \pi'||_\infty < \varepsilon$, q is reachable from q_0 for π', in a given PITA $\mathcal{A}$?

Reachability Analysis

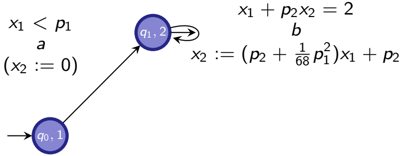
Symbolic class automata for ITA + the first-order theory of reals:

- we build a finite family of class automata
- *PolPar* - a set of polynomials on parameters, depending on guards and updates, defining finite partitions of the set of parameter valuations
- each partition is specified by a satisfiable first-order formula over $(\mathbb{R}, +, \times)$ over parameters
- each class automaton is related to a (non-empty) partition of parameter valuations
- $\{E_k\}_{k \leq n}$ - a family of expressions defining classes, for every level k

A class of a class automaton depends on guards and updates and is defined by:

- a state q
- the values of clocks giving the same ordering of the expressions from E_k

Construction of *PolPar* and $\{E_k\}_{k \le n}$



Initialization:

$PolPar = \emptyset$

$E_1 = \{x_1, 0\}$

$E_2 = \{x_2, 0\}$

Procedure starts from the highest level, $k = 2$

Step 1: consider $C_2 = p_2 x_2 + x_1 - 2$

- compute: $\texttt{lead}(C_2, 2) = p_2$, $\texttt{comp}(C_2, 2) = x_1 - 2$, and $\texttt{compnorm}(C_2, 2) = -\frac{x_1 - 2}{p_2}$
- result: $PolPar = \{p_2\}$ and $E_2 = \{x_2, 0, x_1 - 2, -\frac{x_1 - 2}{p_2}\}$

Step 2a: consider an update of C_2, $x_2 := (p_2 + \frac{1}{68} p_1^2) x_1 + p_2$:

- apply it to every expression of E_2 and add it to E_2
- result: $E_2 = \{x_2, 0, x_1 - 2, -\frac{x_1 - 2}{p_2}, (p_2 + \frac{1}{68} p_1^2) x_1 + p_2\}$

Step 2b: consider an edge between q_0 and q_1:

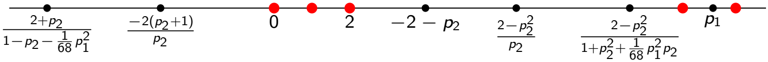
- apply step 1 to differences of any two expressions in E_2 and add it to E_1
- result: $PolPar = \{p_2, p_2 + 1, 1 - p_2 - \frac{1}{68} p_1^2, -p_2^2 - \frac{1}{68} p_1^2 p_2 - 1\}$ and
  $E_1 = \{x_1, 0, 2, -\frac{2(p_2 + 1)}{p_2}, -2 - p_2, \frac{2 + p_2}{1 - p_2 - \frac{1}{68} p_1^2}, \frac{2 - p_2^2}{p_2}, \frac{2 - p_2^2}{1 + p_2^2 + \frac{1}{68} p_1^2 p_2}\}$

Consider the next level, $k = 1$, and repeat the procedure

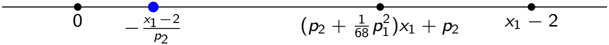
Construction of Class Automata for PITA

Parameter region of *PolPar*: $p_2 < 0, p_2 + 1 < 0, 1 - p_2 - \frac{1}{68}p_1^2 > 0, -1 - p_2^2 - \frac{1}{68}p_1^2 p_2 > 0$

- we obtain the ordering $\preceq_1$ of the expressions in E_1:

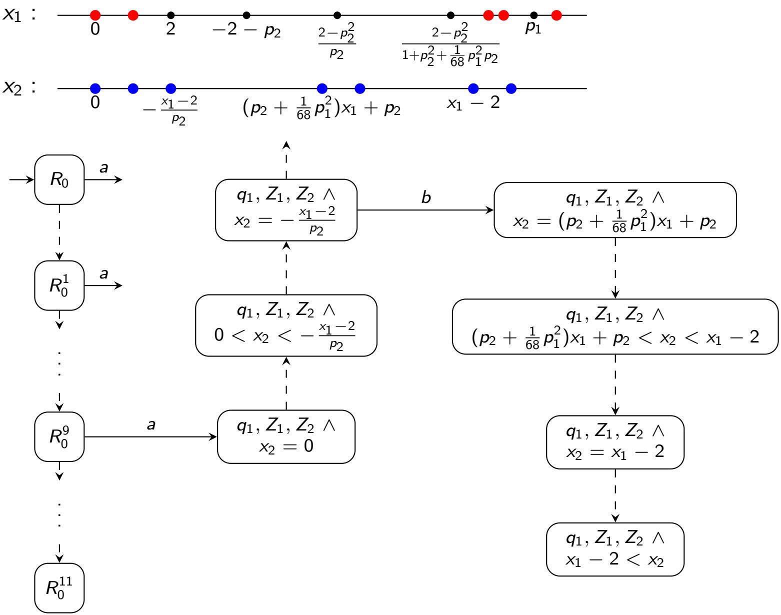


- $R_0 = (q_0, \preceq_1 \wedge x_1 = 0), R_0^1 = (q_0, \preceq_1 \wedge 0 < x_1 < 2)$, $R_0^2 = (q_0, \preceq_1 \wedge x_1 = 2)$, ..., up to $R_0^{11} = (q_0, \preceq_1 \wedge p_1 < x_1)$
- region of *PolPar* and the class from which *a* is fired ($R_0^9 = (q_0, \preceq_1 \frac{2-p_2^2}{1+p_2^2+\frac{1}{68}p_1^2 p_2} \wedge x_1 < p_1)$) determine the ordering of $E_2 \setminus \{x_2\}$:



- transition *b* is fired from the time successor of R_1 for which $x_2 = -\frac{x_1-2}{p_2}$

Construction of Class Automata for PITA



Results

Based on the decidability of the first-order theory of reals and the class automata construction we obtain:

Theorem

The existential, universal and robust reachability problems for PITA are decidable and belong to 2EXPSPACE and PSPACE when the number of clocks is fixed.

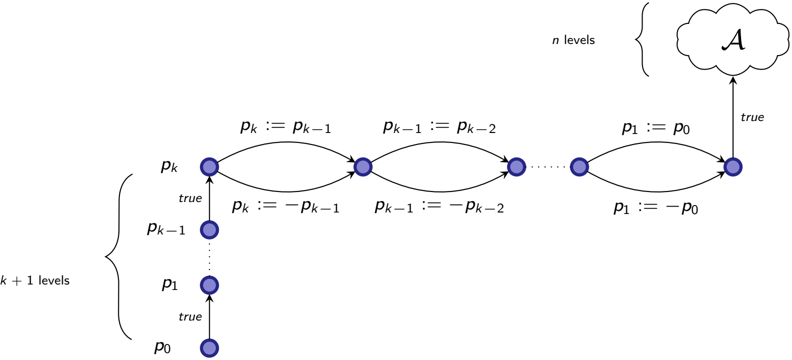
When only additive parametrization is considered, the existential reachability problem *reduces to reachability for ITA*:

Theorem

The existential reachability problem is decidable for additively parametrized PITA, and belongs to 2EXPTIME and PTIME when the number of clocks and parameters is fixed.

Reachability for Additive PITA

- Transform a PITA $\mathcal{A}$ with *n clocks (levels)* and *k parameters* $(p_1, ...p_n)$ into an equivalent ITA $\mathcal{A}'$ with $n+k+1$ *clocks (levels)*.



The reachability problem of additive PITA reduces to the reachability problem of ITA.

Outline

Interrupt Timed Automata (FOSSACS'09, TIME'10, FMSD'12)

Parametric Interrupt Timed Automata (RP'13, FI'16)

Polynomial Interrupt Timed Automata (RP'15, IC'21, IPL'21)

Polynomial ITA (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.

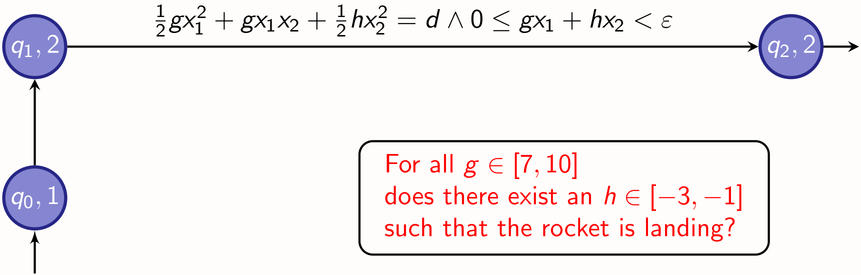
Results

- CTL 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.

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).



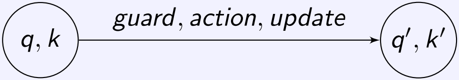
For all $g \in [7, 10]$
does there exist an $h \in [-3, -1]$
such that the rocket is landing?

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

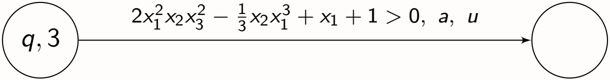
PolITA: syntax

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

- 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.

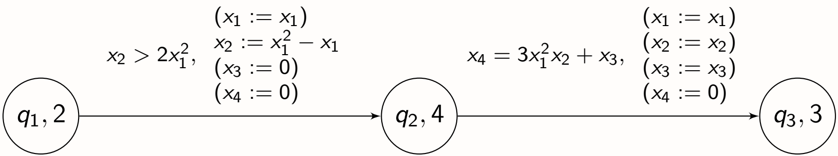


PolITA: updates

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.

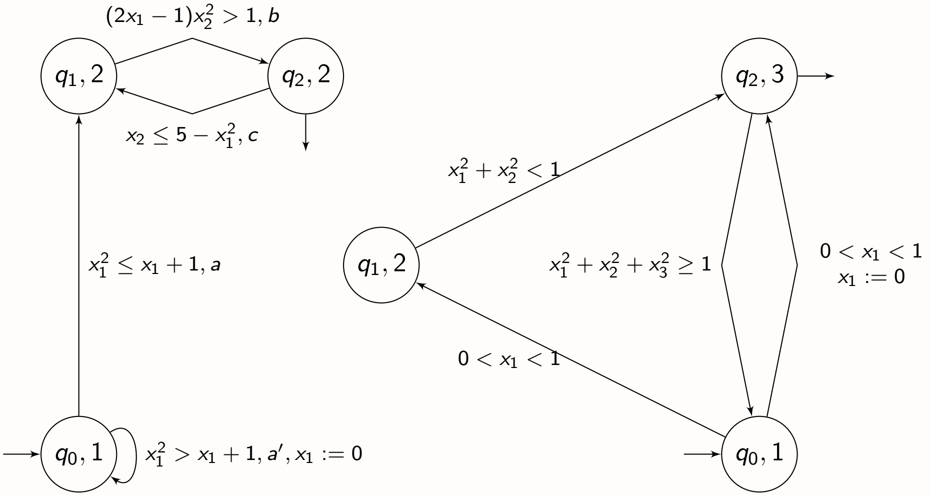


Decreasing level

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

Examples

$\mathcal{A}_2$ in dimension 2 $\qquad$ $\mathcal{A}_3$ in dimension 3



PolITA: semantics

Clock valuation

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

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

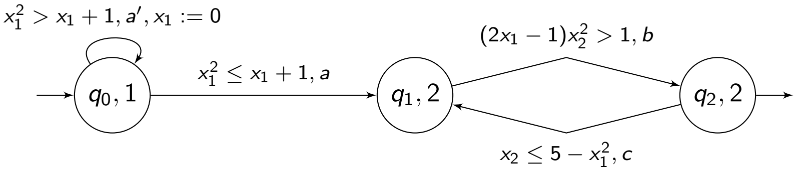
- **configurations** $S = Q \times \mathbb{R}^n$, initial configuration $s_0 = (q_0, v_0)$ with $v_0 = \mathbf{0}$
- **time steps** from q at level k: $(q, v) \xrightarrow{d} (q, v +_k d)$, only x_k is active, with all clock values in $v +_k d$ unchanged except $(v +_k d)(x_k) = v(x_k) + d$
- **discrete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

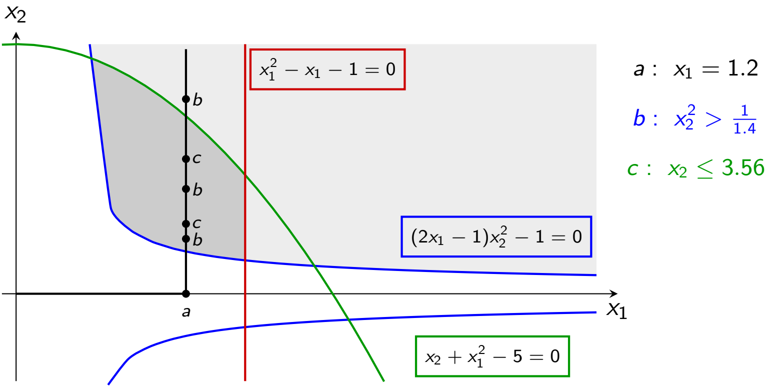
An execution

alternates time and discrete steps

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

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 $-2x^5 + x_1^4 + 20x_1^3 - 10x_1^2 - 50x_1 + 26$.

CTL model-checking

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 $\mathcal{R}_\mathcal{A}$: $(q, C) \rightarrow (q, succ(C))$ 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 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.

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 $\mathcal{R}_{\mathcal{A}}$.

The reachability problem

In a PolITA, a state is reachable if and only if it is reachable in an number of steps exponential w.r.t. the number of clocks and polynomial w.r.t. the number of edges and states.

A non deterministic reachability decision procedure.

- Guess a sequence of transitions;
- Decide the satisfiability problem of a first-order existential formula over the reals.

The reachability problem belongs to EXPSPACE.