Interrupt Timed Automata and their Extensions


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)
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 $x \triangleright c$ 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 $\uparrow$: clocks relevant before can be left unchanged or take an affine expression of clocks of strictly lower level, clocks relevant after are reset;
- level $\downarrow$: clocks relevant after can be left unchanged or take an affine expression of clocks of strictly lower level.
Behaviour of an ITA

level 4

level 3

level 2

level 1

\[
\begin{bmatrix}
    x_1 = 0 \\
    x_2 = 0 \\
    x_3 = 0 \\
    x_4 = 0
\end{bmatrix}
\rightarrow
\begin{bmatrix}
    1.5 \\
    0 \\
    0 \\
    0
\end{bmatrix}
\rightarrow
\begin{bmatrix}
    1.5 \\
    0 \\
    2.1 \\
    0
\end{bmatrix}
\rightarrow
\begin{bmatrix}
    1.5 \\
    0 \\
    2.1 \\
    1.7
\end{bmatrix}
\rightarrow
\begin{bmatrix}
    1.5 \\
    0 \\
    0 \\
    0
\end{bmatrix}
\rightarrow
\begin{bmatrix}
    3.7 \\
    0 \\
    0 \\
    0
\end{bmatrix}
\]
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 \(x_k \otimes \sum_{i < k} a_i x_i + b\) 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

- \( \text{TCTL}^\text{int}_c \) is defined by the following grammar:

\[
\psi ::= p | \psi \land \psi | \neg \psi | \sum_{i \geq 1} a_i \cdot x_i + b \ni 0 | A \psi U \psi | E \psi U \psi
\]

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

**Example.** \( A (x_2 \leq 3) U \text{ safe} \) 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)*

- \( \text{TCTL}_p \) is defined by the following grammar:

\[
\varphi_p ::= p | \varphi_p \land \varphi_p | \neg \varphi_p \quad \text{and} \quad \psi ::= \psi \land \psi | \neg \psi | \varphi_p | A \varphi_p \Upsilon a \varphi_p | E \varphi_p U \Upsilon a \varphi_p
\]

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

**Examples.**

The system is error free for at least 50 t.u. is expressed by \( A (\neg \text{error}) U_{\geq 50} \top \).

The system will reach a safe state within 7 t.u. is expressed by \( A F_{\leq 7} \text{ safe} \).
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 \( \text{ITA}^- \), only the clock of the current level may be updated.

An ITA can be simulated by an \( \text{ITA}^- \) 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 \( \text{ITA}^- \), 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 \( \text{ITA}^- \);
- Guess a sequence of transitions;
- Solve a linear programming problem.

The reachability problem belongs to \( \text{NEXPTIME} \).
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 $C = \sum_{x \in X} a_x x + b$:

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

(a) A PITA $A$ with two interrupt levels

(b) A possible trajectory in $A$

$$
(q_0, 0, 0) \xrightarrow{17} (q_0, 17, 0) \xrightarrow{a} (q_1, 17, 0) \xrightarrow{3} (q_1, 17, 3) \xrightarrow{b} (q_1, 17, 18p_2 + \frac{17}{68} p_1^2)
$$

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 $\pi$ and a real $\varepsilon > 0$ such that for all $\pi'$, with $||\pi - \pi'||_{\infty} < \varepsilon$, $q$ is reachable from $q_0$ for $\pi'$, 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
- \textit{PolPar} - a set of polynomials on parameters, depending on guards and updates, defining \textit{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 \textit{the same ordering} of the expressions from \(E_k\)
Construction of $\text{PolPar}$ and $\{E_k\}_{k\leq n}$

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

Step 1: consider $C_2 = p_2x_2 + x_1 - 2$
- compute: $\text{lead}(C_2, 2) = p_2$, $\text{comp}(C_2, 2) = x_1 - 2$, and $\text{compnorm}(C_2, 2) = -\frac{x_1 - 2}{p_2}$
- result: $\text{PolPar} = \{p_2\}$ 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: $\text{PolPar} = \{p_2, p_2 + 1, 1 - p_2 - \frac{1}{68}p_1^2, -p_2 - \frac{1}{68}p_1^2p_2 - 1\}$ and $E_1 = \{x_1, 0, 2, -\frac{2(p_2+1)}{p_2}, -2 - p_2, \frac{2+p_2}{1-p_2-\frac{1}{68}p_1^2}, \frac{2-p_2}{p_2}, \frac{2-p_2}{1+p_2+\frac{1}{68}p_1^2p_2}\}$

Consider the next level, $k = 1$, and repeat the procedure
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 \( E_1 \):

\[
\begin{align*}
\frac{2+p_2}{1-p_2-\frac{1}{68} p_1^2} & \quad -2(p_2+1) \quad 0 \quad 2 \quad -2 - p_2 \quad \frac{2-p_2^2}{p_2} \quad \frac{2-p_2^2}{1+p_2^2+\frac{1}{68} p_1^2 p_2} \quad p_1
\end{align*}
\]

\( R_0 = (q_0, \preceq_1 \land x_1 = 0), R_0^1 = (q_0, \preceq_1 \land 0 < x_1 < 2), R_0^2 = (q_0, \preceq_1 \land x_1 = 2), \ldots, \) up to \( R_0^{11} = (q_0, \preceq_1 \land p_1 < x_1) \)

region of PolPar and the class from which \( a \) is fired

\( (R_0^9 = (q_0, \preceq_1 \frac{2-p_2^2}{1+p_2^2+\frac{1}{68} p_1^2 p_2} \land x_1 < p_1)) \) determine the ordering of \( E_2 \setminus \{x_2\} \):

\[
\begin{align*}
0 & \quad -\frac{x_1-2}{p_2} & \quad (p_2 + \frac{1}{68} p_1^2) x_1 + p_2 & \quad x_1 - 2
\end{align*}
\]

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

\[ x_1 : \begin{array}{cccccccc}
0 & 2 & -2 - p_2 & \frac{2-p_2^2}{p_2} & \frac{2-p_2^2}{1+p_2^2+168p_1^2p_2} & p_1 \\
\end{array} \]

\[ x_2 : \begin{array}{cccccccc}
0 & \frac{-x_1-2}{p_2} & (p_2 + \frac{1}{68}p_1^2)x_1 + p_2 & x_1 - 2 \\
\end{array} \]

\begin{align*}
R_0 & \xrightarrow{a} q_1, Z_1, Z_2 \land x_2 = -\frac{x_1-2}{p_2} \\
R_0^1 & \xrightarrow{a} q_1, Z_1, Z_2 \land 0 < x_2 < -\frac{x_1-2}{p_2} \\
R_0^9 & \xrightarrow{a} q_1, Z_1, Z_2 \land x_2 = 0 \\
\ldots & \\
R_0^{11} & \\
\end{align*}

\begin{align*}
\xrightarrow{b} q_1, Z_1, Z_2 \land x_2 = (p_2 + \frac{1}{68}p_1^2)x_1 + p_2 \\
\xrightarrow{b} q_1, Z_1, Z_2 \land (p_2 + \frac{1}{68}p_1^2)x_1 + p_2 < x_2 < x_1 - 2 \\
\xrightarrow{b} q_1, Z_1, Z_2 \land x_2 = x_1 - 2 \\
\xrightarrow{b} q_1, Z_1, Z_2 \land x_1 - 2 < x_2 \\
\end{align*}
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 ($p_1, \ldots, p_n$) 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)

- 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 $\varepsilon$).

\[
\frac{1}{2}gx_1^2 + gx_1x_2 + \frac{1}{2}hx_2^2 = d \land 0 \leq gx_1 + hx_2 < \varepsilon
\]

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.
$A = (\Sigma, Q, q_0, X, \lambda, \Delta)$

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

$$\begin{array}{c}
q, k \xrightarrow{\text{guard, action, update}} q', k' \\
\end{array}$$

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

$$\begin{array}{c}
q, 3 \xrightarrow{2x_1^2x_2x_3^2 - \frac{1}{3}x_2x_1^3 + x_1 + 1 > 0, \ a, \ u} \\
\end{array}$$
**PollTA: 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

$A_2$ in dimension 2

$q_1, 2$ to $q_2, 2$

$(2x_1 - 1)x_2^2 > 1, b$

$x_2 \leq 5 - x_1^2, c$

$x_1^2 \leq x_1 + 1, a$

$q_0, 1$ to $q_2, 2$

$q_1, 2$ to $q_2, 3$

$q_0, 1$ to $q_0, 1$

$q_2, 2$ to $q_0, 1$

$q_2, 3$ to $q_0, 1$

$A_3$ in dimension 3

$x_1^2 + x_2^2 < 1$

$q_1, 2$

$x_1^2 + x_2^2 + x_3^2 \geq 1$

$0 < x_1 < 1$

$x_1 := 0$
**PollTA: semantics**

**Clock valuation**

\[ \nu = (\nu(x_1), \ldots, \nu(x_n)) \in \mathbb{R}^n \]

**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 \( s_0 = (q_0, \nu_0) \) with \( \nu_0 = 0 \)
- **time steps** from \( q \) at level \( k \): \( (q, \nu) \xrightarrow{d} (q, \nu + k \cdot d) \), only \( x_k \) is active, with all clock values in \( \nu + k \cdot d \) unchanged except \( (\nu + k \cdot d)(x_k) = \nu(x_k) + d \)
- **discrete steps** \( (q, \nu) \xrightarrow{e} (q', \nu') \) for a transition \( e : q \xrightarrow{g, a, u} q' \) if \( \nu \) satisfies the guard \( g \) and \( \nu' = \nu[u] \).

**An execution**

Alternates time and discrete steps

\[(q_0, \nu_0) \xrightarrow{d_0} (q_0, \nu_0 + \lambda(q_0) \cdot d_0) \xrightarrow{e_0} (q_1, \nu_1) \xrightarrow{d_1} (q_1, \nu_1 + \lambda(q_1) \cdot d_1) \xrightarrow{e_1} \ldots \]
$x_1^2 > x_1 + 1, a', x_1 := 0$

$(2x_1 - 1)x_2^2 > 1, b$

$x_2 \leq 5 - x_1^2, c$

$\begin{align*}
(a) & : x_1 = 1.2 \\
(b) & : x_2^2 > \frac{1}{1.4} \\
(c) & : x_2 \leq 3.56
\end{align*}$

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

$(q_0, 0, 0) \xrightarrow{1.2} (q_0, 1.2, 0) \xrightarrow{a} (q_1, 1.2, 0) \xrightarrow{0.97} (q_1, 1.2, 0.97) \xrightarrow{b} (q_2, 1.2, 0.97) \ldots$
Given $\mathcal{A} = (\Sigma, Q, q_0, X, \lambda, \Delta)$ and $q_f \in Q$

is there an execution from initial configuration $s_0 = (q_0, 0)$ to $(q_f, v)$ for some valuation $v$?

**Build a finite **quotient **automaton** $\mathcal{R}_A$

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

- states of $\mathcal{R}_A$ 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 $\mathcal{R}_A$: $(q, C) \rightarrow (q, \text{succ}(C))$ where $\text{succ}(C)$ is the time successor of $C$, consistent with time elapsing in $\mathcal{T}_A$,
- discrete transitions of $\mathcal{R}_A$: $(q, C) \xrightarrow{e} (q', C')$ for $e : q \xrightarrow{g,a,u} q'$ in $\Delta$ if $C$ satisfies the guard $g$ and $C' = C[u]$, consistent with discrete steps in $\mathcal{T}_A$.

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 $A_2$

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 $R_A$. 
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.