Interrupt Timed Automata

Béatrice Bérard (LIP6), Serge Haddad (LSV)

Séminaire LIAFA, 20 avril 2009

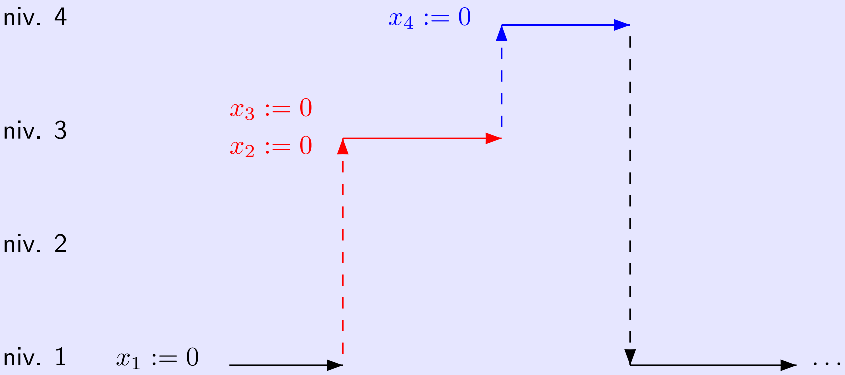
RR LSV 09-01, extended version from Fossacs'09

Motivations

- Theoretical: investigate subclasses of hybrid automata with stopwatches, to obtain decidability results in view of negative results, among them:
  - Henzinger et al. 1998: The reachability problem is decidable for rectangular initalized automata, but becomes undecidable for slight extensions, e.g. adding one stopwatch to timed automata.
  - Cassez, Larsen 2000: Linear hybrid automata and automata with stopwatches (and unobservable delays) are equally expressive.
  - Bouyer, Brihaye, Bruyère, Markey, Raskin 2006: Model checking timed automata with stopwatch observers is undecidable for WCTL (a weighted extension of CTL).
- Practical: Many real-time systems include interruptions (as in processors). An interrupt clock can be seen as a restricted type of stopwatch.

Interruptions and real-time

Several levels with exactly one active clock at each level



Execution : $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{1.5} \begin{bmatrix} 1.5 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{2.1} \begin{bmatrix} 1.5 \\ 0 \\ 2.1 \\ 0 \end{bmatrix} \xrightarrow{1.7} \begin{bmatrix} 1.5 \\ 0 \\ 2.1 \\ 1.7 \end{bmatrix} \xrightarrow{2.2} \begin{bmatrix} 3.7 \\ 0 \\ 2.1 \\ 1.7 \end{bmatrix}$

Outline

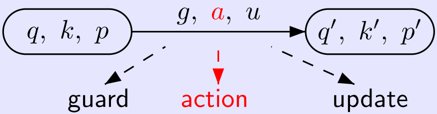
1. ITA model
2. Effective regularity
3. Complexity of the reachability problem
4. Expressiveness
5. Conclusion and perspectives

Outline

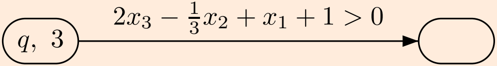
Interrupt Timed Automata (ITA)

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

- The mapping λ associates a level in $\{1, \ldots, n\}$ with each state, $x_{\lambda(q)}$ is the active clock in state q
- The mapping pol associates a timing policy with each state: U for urgent, D for delayed and L for lazy
- Transitions in Δ:



Guard: conjunction of linear constraints on clocks from levels $j \leq k$
$\sum_{j=1}^{k} a_j x_j + b \bowtie 0$, with constants in $\mathbb{Q}$



Updates in ITA

From level k to level k'

Increasing level

Clocks of level greater than k' are unchanged, clocks with level from $k+1$ up to k' are reset, and clocks from level less than or equal to k may be updated by a linear expression $x_i : \sum_{j<i} a_j x_j + b$.

Example

Strictly decreasing level

Clocks of level greater than k' are unchanged and all other clocks (including the one at level k') may be updated by a linear expression $x_i := \sum_{j<i} a_j x_j + b$.

Remark: in a state at level k, all clocks from higher levels are irrelevant.

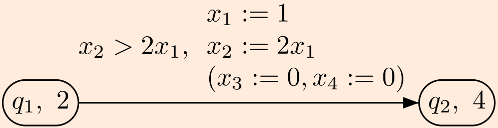
Updates in ITA

From level k to level k'

Increasing level

Clocks of level greater than k' are unchanged, clocks with level from $k+1$ up to k' are reset, and clocks from level less than or equal to k may be updated by a linear expression $x_i : \sum_{j<i} a_j x_j + b$.

Example



Strictly decreasing level

Clocks of level greater than k' are unchanged and all other clocks (including the one at level k') may be updated by a linear expression $x_i := \sum_{j<i} a_j x_j + b$.

Remark: in a state at level k, all clocks from higher levels are irrelevant.

Updates in ITA

From level k to level k'

Increasing level

Clocks of level greater than k' are unchanged, clocks with level from $k+1$ up to k' are reset, and clocks from level less than or equal to k may be updated by a linear expression $x_i : \sum_{j<i} a_j x_j + b$.

Example

$$
\begin{tikzcd}[column sep=huge]
(q_1,\ 2) \arrow[r, "{x_2 > 2x_1,\ \begin{array}{l} x_1 := 1 \\ x_2 := 2x_1 \\ (x_3 := 0, x_4 := 0) \end{array}}"] & (q_2,\ 4) \arrow[r, "{x_3 = 3x_1 + x_2,\ \begin{array}{l} x_1 := 0 \\ x_2 := x_1 + 1, \\ x_3 := 2x_2 \end{array}}"] & (q_3,\ 3)
\end{tikzcd}
$$

Strictly decreasing level

Clocks of level greater than k' are unchanged and all other clocks (including the one at level k') may be updated by a linear expression $x_i := \sum_{j<i} a_j x_j + b$.

Remark: in a state at level k, all clocks from higher levels are irrelevant.

Semantics

For an ITA $\mathcal{A}$

A transition system $\mathcal{T}_{\mathcal{A}} = (S, s_0, \rightarrow)$, with

- configurations $S = \{(q, v, b) \mid q \in Q,\ v \in \mathbb{R}^X, b \in \{\perp, \top\}\}$,
- initial configuration $(q_0, \mathbf{0}, \perp)$,
- transition relation $\rightarrow$

Time step: only the active clock evolves in a state (q, k, p)

- $(q, v, b) \xrightarrow{d} (q, v', \top)$, where $v'(x_k) = v(x_k) + d$ and $v'(x) = v(x)$ for the other clocks.
- If $p = U$, no time step is allowed.

Discrete step:

- $(q, v, b) \xrightarrow{a} (q', v', \perp)$ if there is a transition $q \xrightarrow{\varphi, a, u} q'$ in Δ such that $v \models \varphi$ and $v' = v[u]$.
- If $p = D \wedge b = \perp$, then discrete steps are disallowed.

Language

$\mathcal{L}(\mathcal{A})$ is the set of (finite) timed words associated with a path in $\mathcal{T}_{\mathcal{A}}$ from $(q_0, \mathbf{0})$ to some configuration (q_f, v), for some $q_f \in F$.

ITL : family of languages accepted by ITA.

Semantics

For an ITA $\mathcal{A}$

A transition system $\mathcal{T}_{\mathcal{A}} = (S, s_0, \rightarrow)$, with

- configurations $S = \{(q, v, b) \mid q \in Q,\ v \in \mathbb{R}^X, b \in \{\perp, \top\}\}$,
- initial configuration $(q_0, \mathbf{0}, \perp)$,
- transition relation $\rightarrow$

Time step: only the active clock evolves in a state (q, k, p)

- $(q, v, b) \xrightarrow{d} (q, v', \top)$, where $v'(x_k) = v(x_k) + d$ and $v'(x) = v(x)$ for the other clocks.
- If $p = U$, no time step is allowed.

Discrete step:

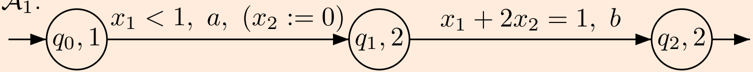
- $(q, v, b) \xrightarrow{a} (q', v', \perp)$ if there is a transition $q \xrightarrow{\varphi, a, u} q'$ in Δ such that $v \models \varphi$ and $v' = v[u]$.
- If $p = D \wedge b = \perp$, then discrete steps are disallowed.

Language

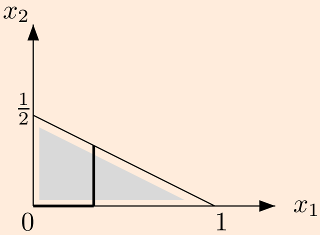
$\mathcal{L}(\mathcal{A})$ is the set of (finite) timed words associated with a path in $\mathcal{T}_{\mathcal{A}}$ from $(q_0, \mathbf{0})$ to some configuration (q_f, v), for some $q_f \in F$.
ITL : family of languages accepted by ITA.

Examples

$\mathcal{A}_1$:



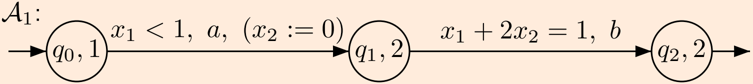
accepts $L_1 = \{(a, 1-\tau)(b, 1-\tau/2) \mid 0 < \tau \leq 1\}$, with trajectories in:



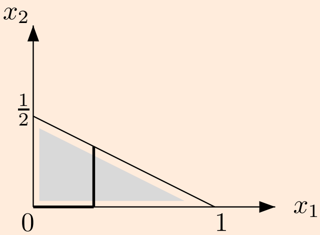
Light gray zone for state q_1:

$(0 < x_1 < 1,\ 0 < x_2 < -\frac{1}{2}x_1 + \frac{1}{2})$

Examples

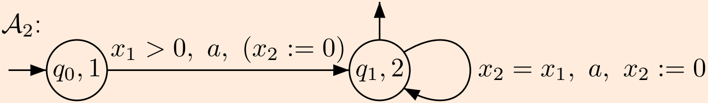


accepts $L_1 = \{(a, 1-\tau)(b, 1-\tau/2) \mid 0 < \tau \leq 1\}$, with trajectories in:



Light gray zone for state q_1:

$(0 < x_1 < 1,\ 0 < x_2 < -\frac{1}{2}x_1 + \frac{1}{2})$



accepts $L_2 = \{(a, \tau)(a, 2\tau) \ldots (a, n\tau) \mid n \in \mathbb{N}, \tau > 0\}$.

Outline

ITA model

2 Effective regularity

Complexity of the reachability problem

Expressiveness

Conclusion and perspectives

A generalized region automaton

Theorem

For a language L in ITL, $Untime(L)$ is effectively regular.

Principle: For an ITA $\mathcal{A} = (\Sigma, X, Q, q_0, F, \lambda, pol, \Delta)$

A finite set $Exp(q)$ of linear expressions is associated with each state $q \in Q$. $Exp(q) = \bigcup_{k \leq \lambda(q)} E_k$, where the sets $E_k = \{0, x_k\}$ are obtained iteratively downward:

- adding the *complements* of x_k in guards from level k,
- saturating E_k by applying updates of appropriate transitions to expressions of E_k,
- saturating E_j $(j < k)$ by applying updates of appropriate transitions to differences of expressions of E_k.

Two valuations are equivalent in state q with level k if they produce the same preorders for linear expressions in each E_i, $i \leq k$.

- A class is a pair $R = (q, \{\preceq_k\}_{k \leq \lambda(q)})$ where $\preceq_k$ is a total preorder on E_k.
- Time successors $R \to R'$ and discrete steps $R \xrightarrow{a} R'$ are then defined.

A generalized region automaton

Theorem

For a language L in ITL, $Untime(L)$ is effectively regular.

Principle: For an ITA $\mathcal{A} = (\Sigma, X, Q, q_0, F, \lambda, pol, \Delta)$

A finite set $Exp(q)$ of linear expressions is associated with each state $q \in Q$.
$Exp(q) = \bigcup_{k \leq \lambda(q)} E_k$, where the sets $E_k = \{0, x_k\}$ are obtained iteratively downward:

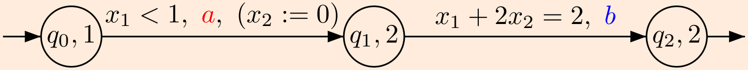
- adding the *complements* of x_k in guards from level k,
- saturating E_k by applying updates of appropriate transitions to expressions of E_k,
- saturating E_j $(j < k)$ by applying updates of appropriate transitions to differences of expressions of E_k.

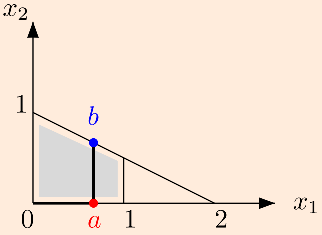
Two valuations are equivalent in state q with level k if they produce the same preorders for linear expressions in each E_i, $i \leq k$.

- A class is a pair $R = (q, \{\preceq_k\}_{k \leq \lambda(q)})$ where $\preceq_k$ is a total preorder on E_k.
- Time successors $R \rightarrow R'$ and discrete steps $R \xrightarrow{a} R'$ are then defined.

Example

For automaton $\mathcal{A}_3$





$E_1 = \{x_1, 0, 1, 2\}$ and $E_2 = \{x_2, 0, -\frac{1}{2}x_1 + 1\}$

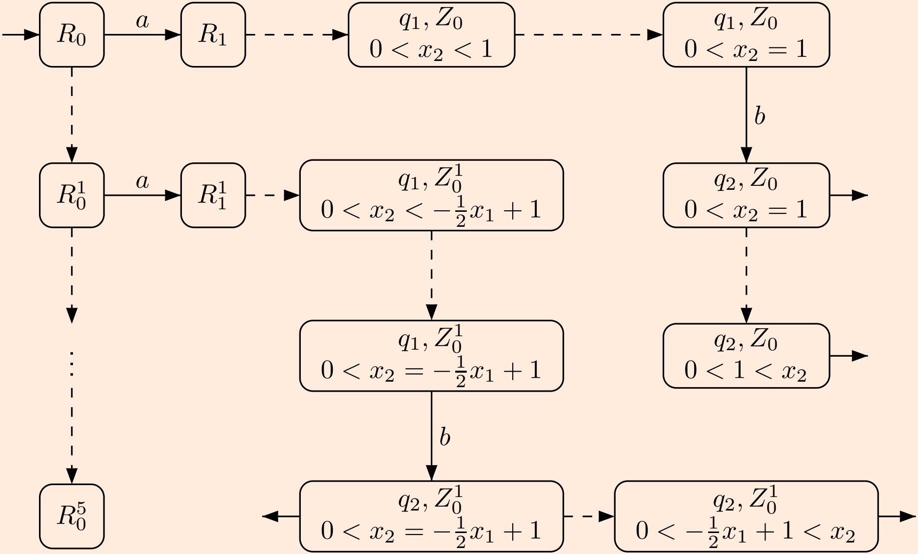
$R_0 = (q_0, Z_0)$ with $Z_0 : x_1 = 0 < 1 < 2$

Time successors of R_0 are $R_0^i = (q_0, Z_0^i)$ with:
$Z_0^1 = (0 < x_1 < 1 < 2)$, $Z_0^2 = (0 < x_1 = 1 < 2)$, $Z_0^3 = (0 < 1 < x_1 < 2)$, $Z_0^4 = (0 < 1 < x_1 = 2)$ and $Z_0^5 = (0 < 1 < 2 < x_1)$

Discrete transitions with action a : $R_0 \xrightarrow{a} R_1 = (q_1, Z_0, x_2 = 0 < \frac{1}{2})$, since $x_1 = 0$, and $R_0^1 \xrightarrow{a} R_1^1 = (q_1, Z_0^1, x_2 = 0 < -\frac{1}{2}x_1 + 1)$

Discrete transitions with action b : from classes such that $x_2 = -\frac{1}{2}x_1 + 1$.

Example (cont.)



Outline

ITA_{-} and reachability

An elementary path in the previous graph can be non deterministically guessed in 2-EXPSPACE leading to the decidability of reachability.

The subclass ITA_{-}

An ITA_{-} is an ITA where updates are restricted to transitions increasing the level, only for the current clock (apart from initializations).

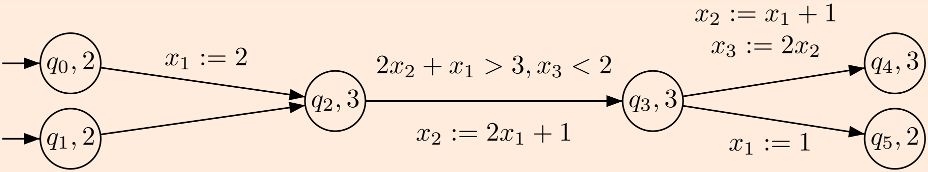
- Reachability in ITA_{-} is decidable in NEXPTIME (existence of an exponentially bounded path).
- An ITA can be transformed into an doubly exponentially larger ITA_{-} with the same clocks accepting the same language.
- Reachability in ITA is decidable in 2-NEXPTIME by combination of these results.
- When the number of clocks is fixed, the reachability problem is NP.

From ITA to ITA_

Principle: Record the forbidden resets in the states

Apply them when needed and use urgent state copies to decrease level.

Example

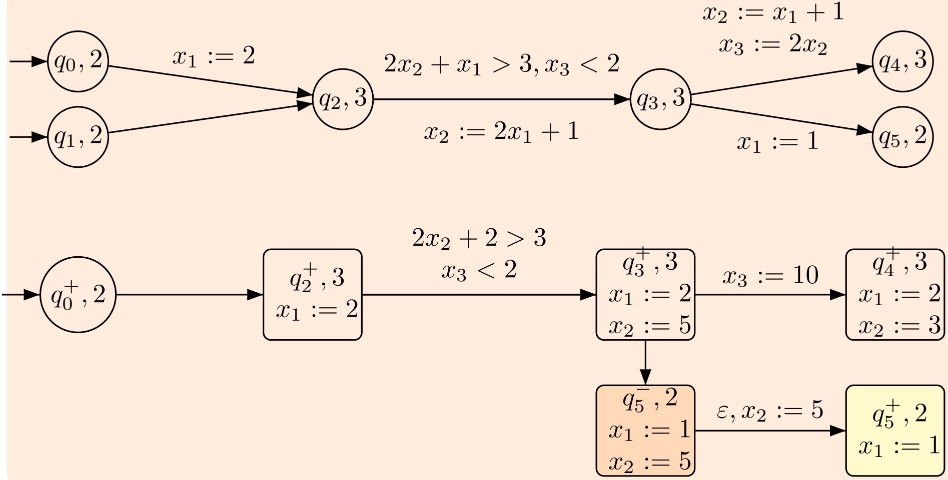


From ITA to ITA$_{-}$

Principle: Record the forbidden resets in the states

Apply them when needed and use urgent state copies to decrease level.

Example

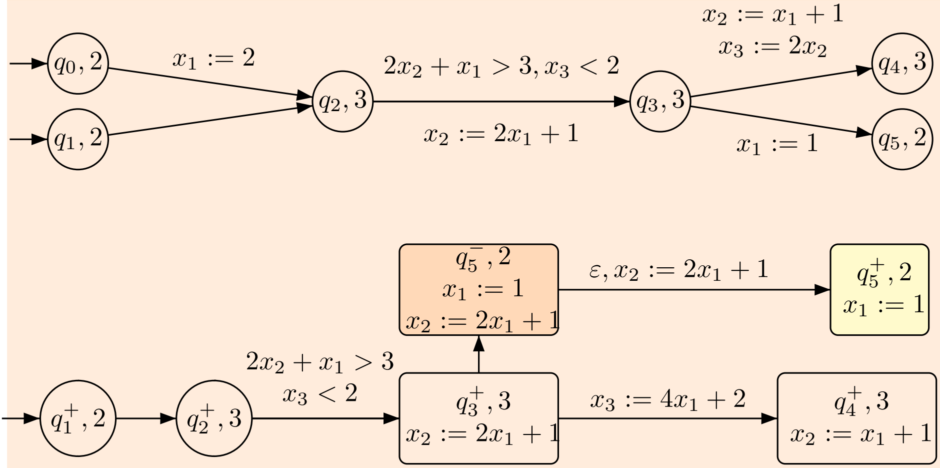


From ITA to ITA$_-$

Principle: Record the forbidden resets in the states

Apply them when needed and use urgent state copies to decrease level.

Example

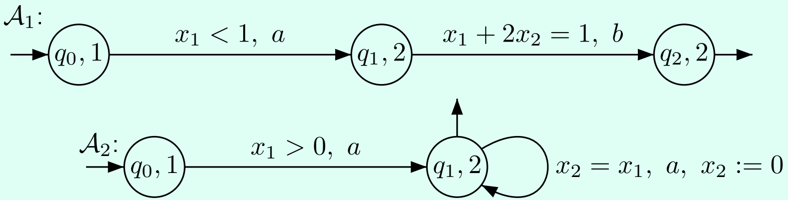


Outline

ITL is neither contained in TL nor in CRTL

Comparing with TA (Timed Automata)

There is no timed automaton accepting L_1 or L_2.



Comparing with CRTA (Controlled Real-Time Automata)

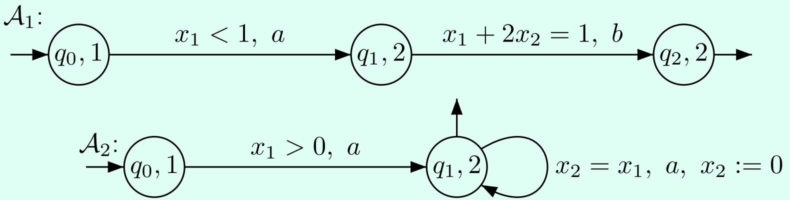
Demichelis, Zielonka, 1998:

There is no controlled real-time automaton accepting L_2.

ITL is neither contained in TL nor in CRTL

Comparing with TA (Timed Automata)

There is no timed automaton accepting L_1 or L_2.



Comparing with CRTA (Controlled Real-Time Automata)

Demichelis, Zielonka, 1998:

There is no controlled real-time automaton accepting L_2.

TL is not contained in ITL

A pumping Lemma

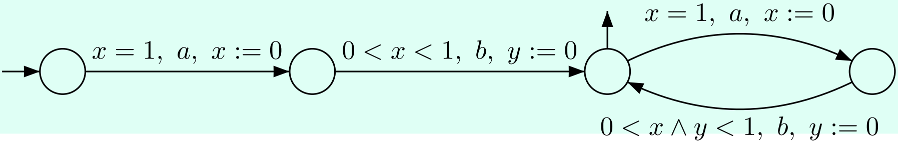
For a language $\mathcal{L}$ in ITL, there exists $B \in \mathbb{N}$ s.t.
given any word (with strictly increasing dates) belonging to $\mathcal{L}$ with B consecutive positions , there are two (**possibly equal**) positions s.t. the subword between these positions can be

- Either duplicated with a **non null** time shift greater or equal than its duration,
- Or erased **without** time shift (in this case the subword is non empty)

and the new word still belongs to $\mathcal{L}$.

ITL and TL are incomparable

A language L_3 accepted by a TA but not by any ITA:



Closure results

ITL is not closed under complement: L_3^c is in ITL

ITL is not closed under intersection : L_3 is the intersection of the two following ITL

- $(a, 1)(b, \tau_1) \ldots (a, n)(b, \tau_n)$ with $\forall 1 \leq i \leq n \; i < \tau_i < i + 1$
- $(a, \tau'_1)(b, \tau_1) \ldots (a, \tau'_n)(b, \tau_n)$ with $\forall 1 \leq i \leq n - 1 \; \tau_{i+1} - \tau_i < 1$

Combining ITA and CRTA

into ITA^+

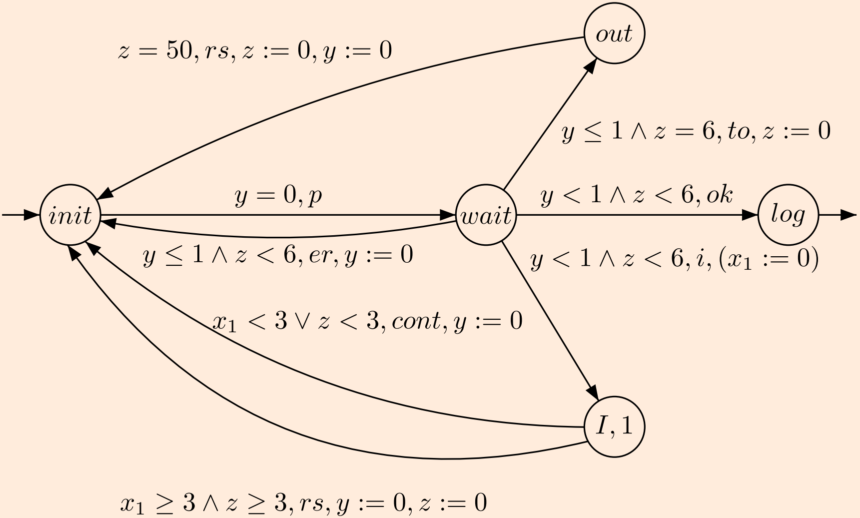
- A set Q of states with either a color or a level, and a velocity.
- A set X of interrupt clocks and a set Y of clocks with the features of CRTA clocks: color, lower and upper bound.
  Clocks with the color of the state are active in this state with same velocity.
  Exactly one interrupt clock active in states with a level.
- guards are of the form $\varphi_1 \wedge \varphi_2$, with φ_1 a guard on X and φ_2 a guard on Y, with the constraints of their respective models,
- updates are of the form $u_1 \wedge u_2$, with u_1 an update on X and u_2 an update on Y, also with the constraints of their respective models.

Reachability

The reachability problem remains decidable in the class ITA^+.
It belongs to 2-NEXPTIME (NEXPTIME with ITA_-^+) and is PSPACE-complete when the number of interrupt clocks is fixed.

An example of ITA$^+$

A login procedure



Outline

Conclusion and perspectives

Summary of results

- An appropriate model for a frequent pattern of discrete-event systems.
- Decidability of the reachability problem and contrasting complexity results.
- Incomparability with TA motivating a “decidable” combination of models.

Perspectives

- Lower bounds for the reachability problem.
- Model-checking ITA.