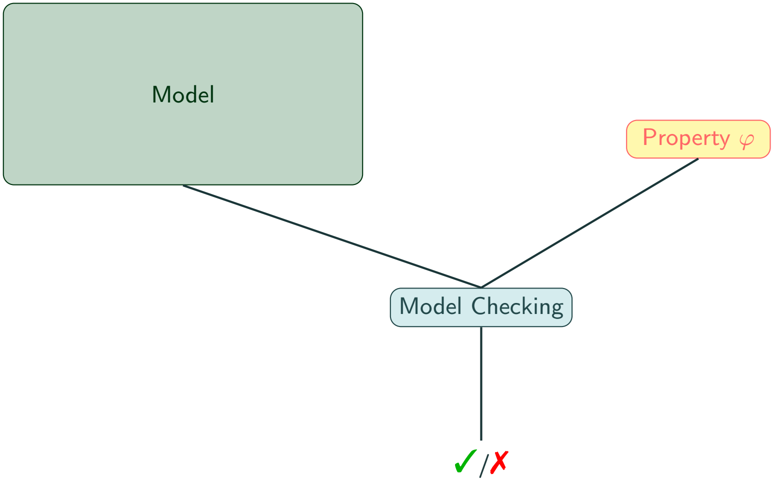
Higher-Dimensional Timed Automata

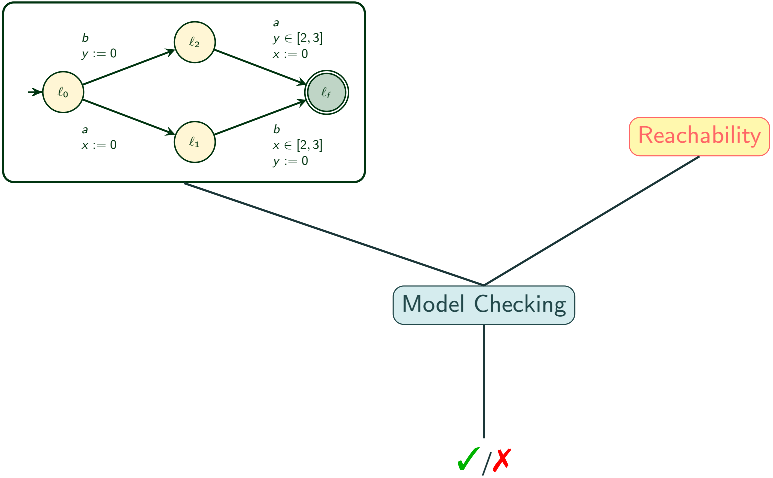
Amazigh Amrane [2] *Hugo Bazille* [2] ***Emily Clement*** [1] *Uli Fahrenberg* [2]

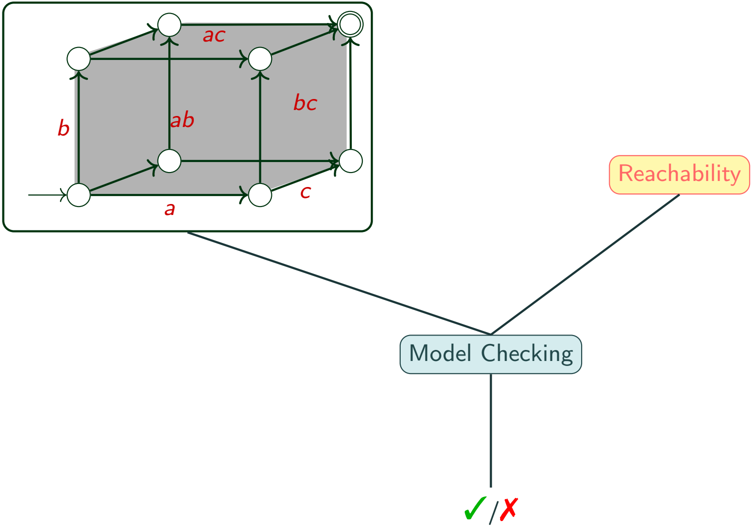
[1]CNRS, LIPN UMR 7030, Université Sorbonne Paris Nord, F-93430 Villetaneuse, France
[2]EPITA Research Laboratory (LRE), Paris, France

7th of November 2024

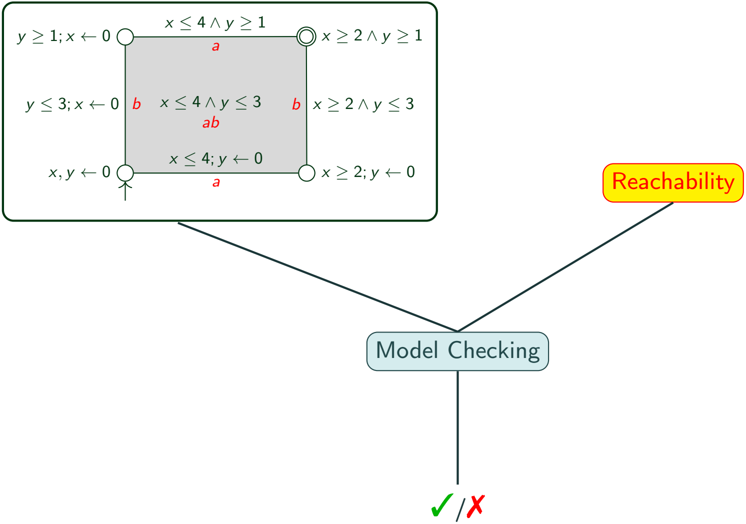
Higher Dimensional (Timed) Automata and Model Checking



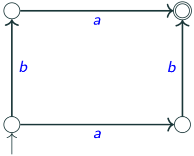

b
y := 0
a
y ∈ [2, 3]
x := 0
ℓ0
ℓ2
ℓ1
ℓf
a
x := 0
b
x ∈ [2, 3]
y := 0
Reachability
Model Checking
✓/✗

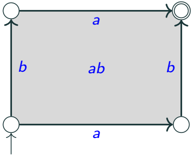

ac
bc
b
ab
c
a
Reachability
Model Checking
✓/✗

Higher Dimensional (Timed) Automata and Model Checking



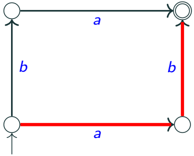
- **Goal** : represent non-interleaving concurrency : $a||b \neq a.b + b.a$
- Higher dimensional Automata of dimension 1 ($\mathcal{A}_1$, left), and dimension 2 ($\mathcal{A}_2$, right) :

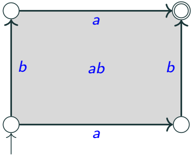


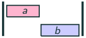


Example :

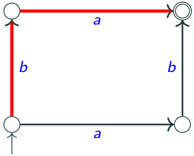
- **Goal** : represent non-interleaving concurrency : $a||b \neq a.b + b.a$
- Higher dimensional Automata of dimension 1 ($\mathcal{A}_1$, left), and dimension 2 ($\mathcal{A}_2$, right) :

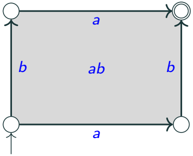




Example : 

- **Goal** : represent non-interleaving concurrency : $a||b \neq a.b + b.a$
- Higher dimensional Automata of dimension 1 ($\mathcal{A}_1$, left), and dimension 2 ($\mathcal{A}_2$, right) :



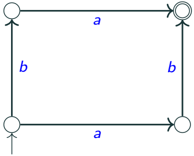


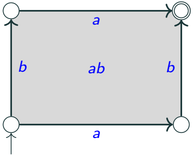
Example :

- Language of the HDA :

$$L(\mathcal{A}_1) = \{ (\ a \rightarrow b \ ) , (\ b \rightarrow a \ ) \}$$

- **Goal** : represent non-interleaving concurrency : $a||b \neq a.b + b.a$
- Higher dimensional Automata of dimension 1 ($\mathcal{A}_1$, left), and dimension 2 ($\mathcal{A}_2$, right) :



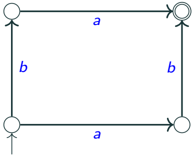


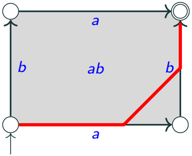
Example :

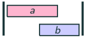
- Language of the HDA :

- **Goal** : represent non-interleaving concurrency : $a||b \neq a.b + b.a$

- Higher dimensional Automata of dimension 1 ($\mathcal{A}_1$, left), and dimension 2 ($\mathcal{A}_2$, right) :



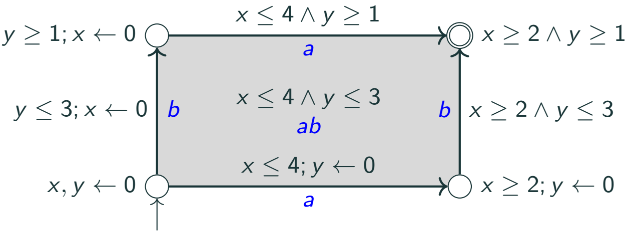


Example : 

- Language of the HDA :

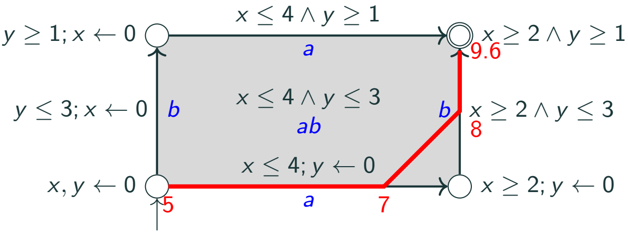
$$L(\mathcal{A}_2) = \left\{ \begin{pmatrix} a \\ b \end{pmatrix}, \left(a \to b \right), \left(b \to a \right) \right\}$$

Higher Dimensional Timed Automata [1] (HDTA)



1. Fahrenberg, « Higher-Dimensional Timed and Hybrid Automata », 2022.

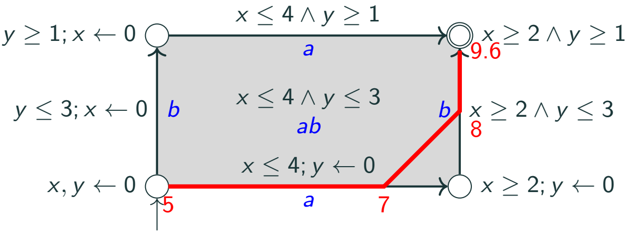
Higher Dimensional Timed Automata [1] (HDTA)



- Event/Transition can have a **duration**
- **Events** can occur **simultaneously**.

1. Fahrenberg, « Higher-Dimensional Timed and Hybrid Automata », 2022.

Higher Dimensional Timed Automata [1] (HDTA)



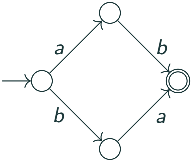
- Event/Transition can have a **duration**
- **Events** can occur **simultaneously**.

Our Contribution

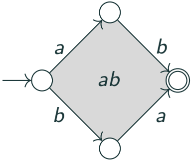
- Express the **language of HDTA**.
- Explain the **links between HDA, TA and HDTA**.
- Extend some decidability/undecidability **TA results** to HDTA.

1. Fahrenberg, « Higher-Dimensional Timed and Hybrid Automata », 2022.

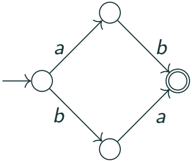
- Two-events HDA
  - ▷ $a.b + b.a$:



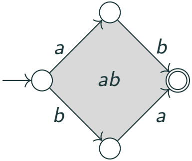
  - ▷ $a||b$:



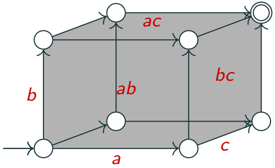
- Two-events HDA
  - $a.b + b.a$:



  - $a||b$:

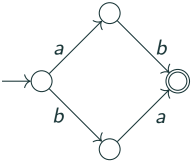


- Three-events HDA
  - $a||b + b||c + a||c$:

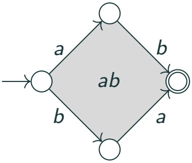


Examples of traces :

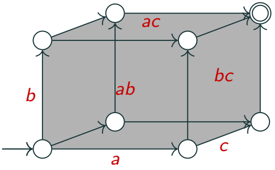
- Two-events HDA
  - ▷ $a.b + b.a$:



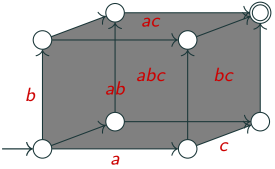
  - ▷ $a||b$:



- Three-events HDA
  - ▷ $a||b + b||c + a||c$:



  - ▷ $a||b||c$:



Examples of traces :

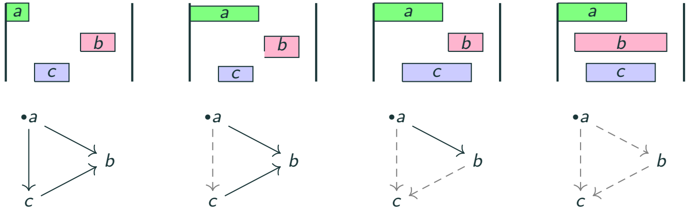
Events representation : interval pomset with interfaces (iiPomset)

- Two partial order events
  - ▷ $<$: precedence order (rep with $\longrightarrow$) , $\dashrightarrow$: event order.
  - ▷ $< \cup \dashrightarrow$: **total** relation.

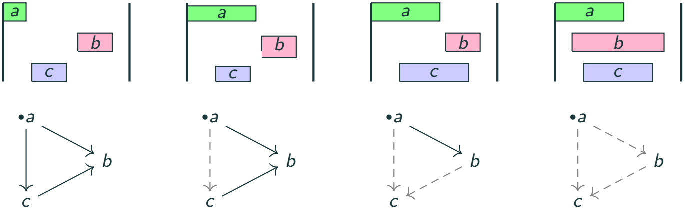
Events representation : interval pomset with interfaces (iiPomset)

- Two partial order events
  - ▷ $<$: precedence order (rep with $\longrightarrow$) , $\dashrightarrow$: event order.
  - ▷ $< \cup \dashrightarrow$: **total** relation.
- Interfaces
  - ▷ **Source/Target interfaces** : S/T : $<$-minimal/maximal.

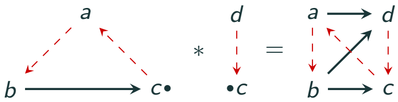
- Two partial order events
  - ▷ $<$: precedence order (rep with $\longrightarrow$) , $\dashrightarrow$: event order.
  - ▷ $< \cup \dashrightarrow$: **total** relation.
- Interfaces
  - ▷ **Source/Target interfaces** : S/T : $<$-minimal/maximal.
- Representation of events as interval



- Two partial order events
  - ▷ $<$: precedence order (rep with $\longrightarrow$) , $\dashrightarrow$: event order.
  - ▷ $< \cup \dashrightarrow$: **total** relation.
- Interfaces
  - ▷ **Source/Target interfaces** : S/T : $<$-minimal/maximal.
- Representation of events as interval



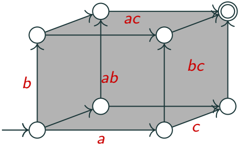
- Gluing composition :



Language of HDA

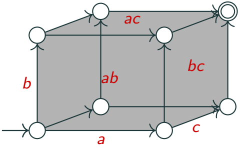
- Example of languages

  ▷ A_1 :



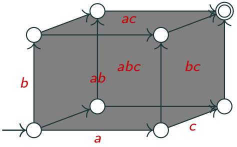
$$L_1 = \{abc, acb, bac, bca, cab, cba\} \cup \left\{ \begin{pmatrix} a \\ b \to c \end{pmatrix}, \begin{pmatrix} a \\ c \to b \end{pmatrix}, \begin{pmatrix} b \\ a \to c \end{pmatrix}, \begin{pmatrix} b \\ a \to c \end{pmatrix}, \begin{pmatrix} b \\ c \to a \end{pmatrix}, \begin{pmatrix} c \\ a \to b \end{pmatrix}, \begin{pmatrix} c \\ b \to a \end{pmatrix} \right\}$$

- Example of languages
  - A_1 :



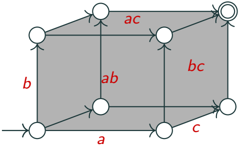
$$L_1 = \{abc, acb, bac, bca, cab, cba\} \cup \left\{ \begin{pmatrix} a \\ b \to c \end{pmatrix}, \begin{pmatrix} a \\ c \to b \end{pmatrix}, \begin{pmatrix} b \\ a \to c \end{pmatrix}, \begin{pmatrix} b \\ a \to c \end{pmatrix}, \begin{pmatrix} b \\ c \to a \end{pmatrix}, \begin{pmatrix} c \\ a \to b \end{pmatrix}, \begin{pmatrix} c \\ b \to a \end{pmatrix} \right\}$$

  - A_2 :

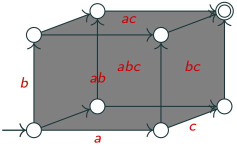


$$: L_2 = L_1 \cup \left\{ \begin{pmatrix} a \\ b \\ c \end{pmatrix} \right\}$$

- Example of languages

  ▷ A_1 : 

  $$L_1 = \{abc, acb, bac, bca, cab, cba\} \cup \left\{ \begin{pmatrix} a \\ b \to c \end{pmatrix}, \begin{pmatrix} a \\ c \to b \end{pmatrix}, \begin{pmatrix} b \\ a \to c \end{pmatrix}, \begin{pmatrix} b \\ a \to c \end{pmatrix}, \begin{pmatrix} b \\ c \to a \end{pmatrix}, \begin{pmatrix} c \\ a \to b \end{pmatrix}, \begin{pmatrix} c \\ b \to a \end{pmatrix} \right\}$$

  ▷ A_2 : 

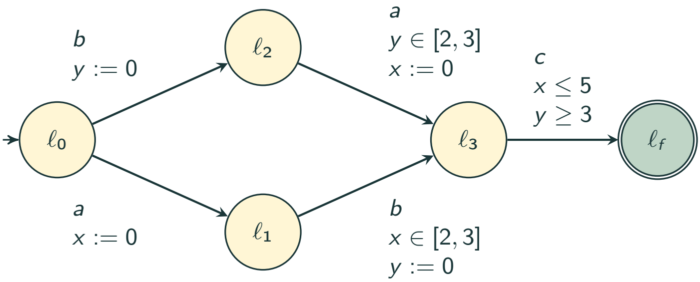
  $$: L_2 = L_1 \cup \left\{ \begin{pmatrix} a \\ b \\ c \end{pmatrix} \right\}$$

- The language of an HDA $A = (X, X_\bot, X_\top)$ is :

$$\boldsymbol{L(A) = \{ev(\alpha) | \alpha \text{ accepting path in } X\}}$$

- Example of Scheduling of events a, b, c

Time constraints impose that between event a and b, at least (*resp.* at most) 2 (*resp.* 3) time units elapses

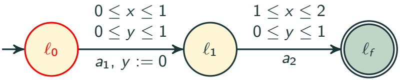


- Semantics of transitions
  - Delay transitions $(\ell, v) \xrightarrow{\delta} (\ell, v + \delta)$
  - Action transitions : $(\ell, v) \xrightarrow{a_1} (\ell_1, v[y := 0])$

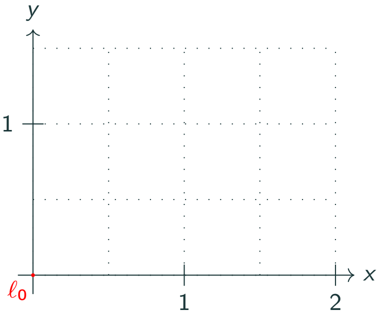
2. Alur et Dill, « A Theory of Timed Automata », 1994.

Clocks evolution example

- Timed automaton $\mathcal{A}$:

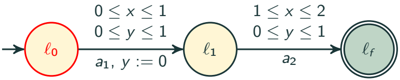


- Evolution of clocks x and y during the run

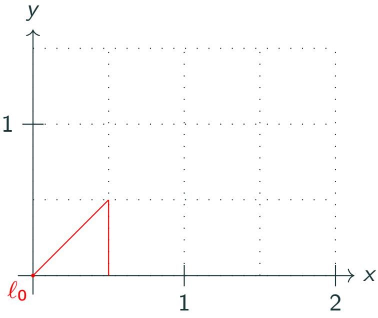


Clocks evolution example

- Timed automaton $\mathcal{A}$:

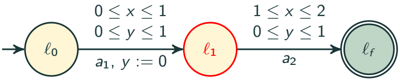


- Evolution of clocks x and y during the run

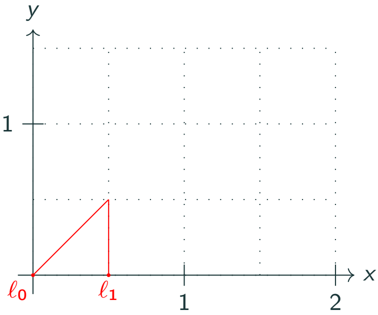


Clocks evolution example

- Timed automaton $\mathcal{A}$:

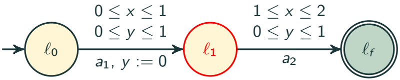


- Evolution of clocks x and y during the run

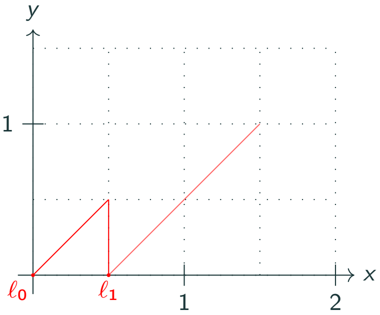


Clocks evolution example

- Timed automaton $\mathcal{A}$:

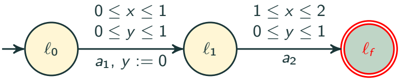


- Evolution of clocks x and y during the run

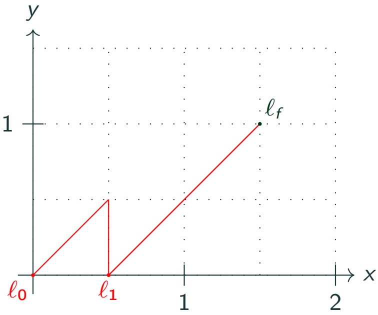


Clocks evolution example

- Timed automaton $\mathcal{A}$:



- Evolution of clocks x and y during the run

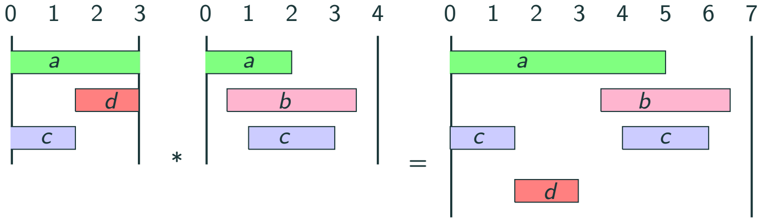


- Timed ipomsets is composed of :
  - ▷ An ipomset
  - ▷ A duration d
  - ▷ A map σ labelling all events to time intervals

- Ipomsets (left), Timed ipomsets (right)

  - ▷ Starter : x_1, x_3 of respective label a and c
  - ▷ Target : x_2 of label b
  - ▷ $\sigma(a) = (0, 3), \sigma(b) = (0, 1.5), \sigma(c) = (1.5, 3)$
  - ▷ Total duration $d = 3$

Gluing on Timed Ipomsets

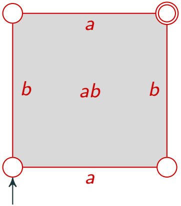


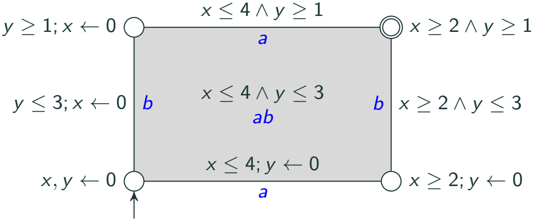
- Definition :

A HDTA is a tuple $(X, X_\perp, X_\top, \lambda, \mathcal{C}, \text{inv}, \text{exit})$ where :

▷ $(\boldsymbol{X}, \boldsymbol{X}_\perp, \boldsymbol{X}_\top, \boldsymbol{\lambda})$ is an HDA

- Example with events a and b : HDA (left) of the HDTA (right)



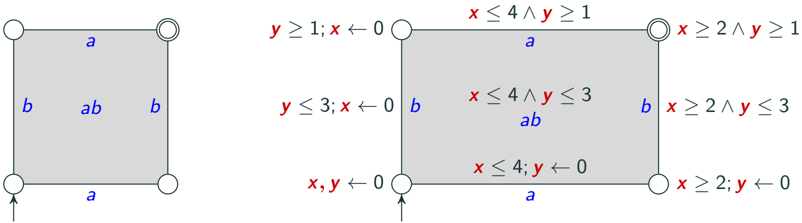


- Definition :

A HDTA is a tuple $(X, X_\perp, X_\top, \lambda, \mathcal{C}, \text{inv}, \text{exit})$ where :

- ▷ $(X, X_\perp, X_\top, \lambda)$ is an HDA
- ▷ $\mathcal{C}$: set of clocks

- Example with events a and b : HDA (left) of the HDTA (right)

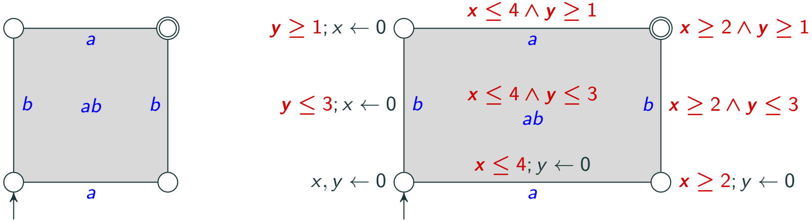


- Definition :

A HDTA is a tuple $(X, X_\perp, X_\top, \lambda, \mathcal{C}, \text{inv}, \text{exit})$ where :

- $(X, X_\perp, X_\top, \lambda)$ is an HDA
- $\mathcal{C}$: set of clocks
- **inv : $X \to \phi(\mathcal{C})$** assign **invariant conditions** to cells.

- Example with events a and b : HDA (left) of the HDTA (right)

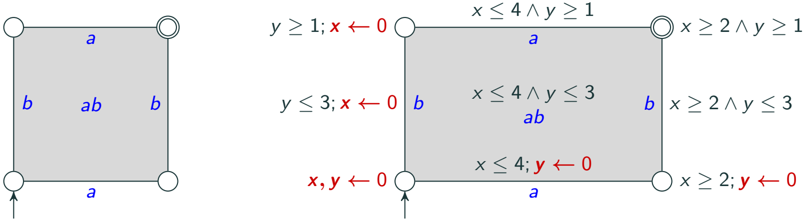


- Definition :

A HDTA is a tuple $(X, X_\perp, X_\top, \lambda, \mathcal{C}, \text{inv}, \text{exit})$ where :

- ▷ $(X, X_\perp, X_\top, \lambda)$ is an HDA
- ▷ $\mathcal{C}$: set of clocks
- ▷ inv : $X \to \phi(\mathcal{C})$ assign invariant conditions to cells.
- ▷ exit : $\boldsymbol{X} \to 2^{\boldsymbol{\mathcal{C}}}$ assign **exit conditions** to cells.

- Example with events a and b : HDA (left) of the HDTA (right)



Quizz : suppose that a and b are not in concurrency

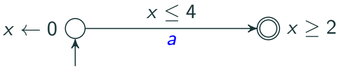
Let us draw the HDTA of $a : [2, 4]$ and $b : [1, 3]$ separately :

Timing duration of events :

- ▷ $a : [2, 4]$ time units
- ▷ $b : [1, 3]$ time units

Quizz : suppose that a and b are not in concurrency

Let us draw the HDTA of $a : [2, 4]$ and $b : [1, 3]$ separately :

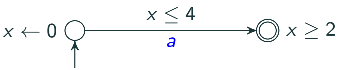


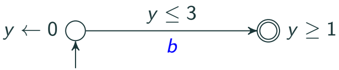
Timing duration of events :

- $a : [2, 4]$ time units
- $b : [1, 3]$ time units

Quizz : suppose that a and b are not in concurrency

Let us draw the HDTA of $a : [2,4]$ and $b : [1,3]$ separately :





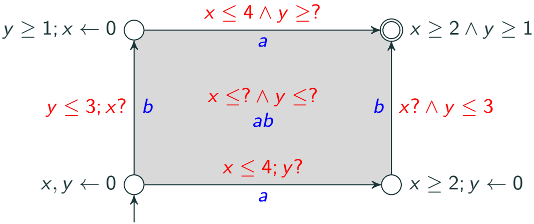
Timing duration of events :

- $a : [2,4]$ time units
- $b : [1,3]$ time units

Quizz : suppose that a and b are not in concurrency

Let us draw the HDTA of $a : [2,4]$ and $b : [1,3]$ separately :

- Let's put them together



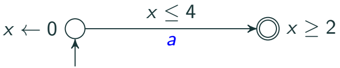
Timing duration of events :

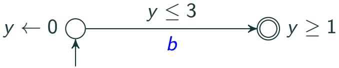
- $a : [2,4]$ time units
- $b : [1,3]$ time units

Examples of HDTA

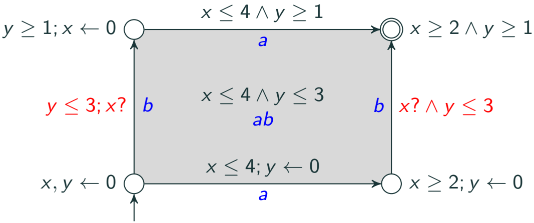
Quizz : suppose that a and b are not in concurrency

Let us draw the HDTA of $a : [2, 4]$ and $b : [1, 3]$ separately :





- Let's put them together



Timing duration of events :

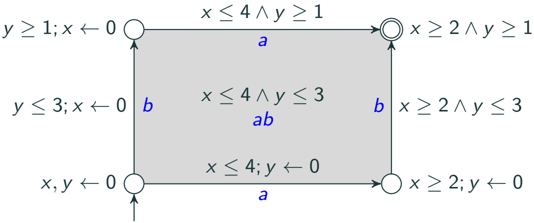
- ▷ $a : [2, 4]$ time units
- ▷ $b : [1, 3]$ time units

Examples of HDTA

Quizz : suppose that a and b are not in concurrency

Let us draw the HDTA of $a : [2,4]$ and $b : [1,3]$ separately :

- Let's put them together

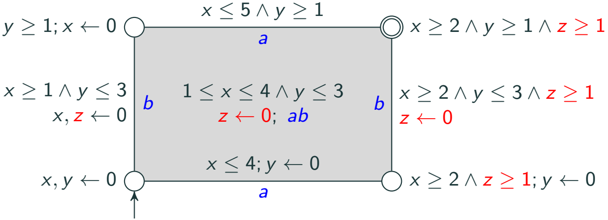


Timing duration of events :

- $a : [2,4]$ time units
- $b : [1,3]$ time units

Example of HDTA

- Second example : adding timing constraints between events...



Timing duration of events :

- ▷ a : $[2, 4]$ time units
- ▷ b : $[1, 3]$ time units

Constraints between starting/ending dates

- ▷ 1 time unit should elapse between b's starting date and a's starting date
- ▷ 1 time unit should elapse between b's ending date and a's ending date

Differences between TA and HDTA

- Cells
  - ▷ 0-cells : location,
  - ▷ $1-$cell : edges,
  - ▷ $d-$cell, $d > 1$.

- Differences

	TA	HDTA
Difference between locations, edges	Yes	No
Exit conditions	Edges	On $d-$cells, $\forall d$
Invariants	Locations	On $d-$cells, $\forall d$
Reset	Edges	On $d-$cells, $\forall d$
Events	Instantaneous	With duration
Concurrency	Interleaving	Possibly simultaneous

Interval Delay words, untimed of Timed Ipomsets

- Timed Ipomsets and Interval delay words
  - Timed Ipomsets : (P, σ_P, d_P).

Interval Delay words, untimed of Timed Ipomsets

- Timed Ipomsets and Interval delay words
  - Timed Ipomsets : (P, σ_P, d_P).
  - Steps sequence (HDA)

$$(S_{Q_0}, Q_0, T_{Q_0}) * (S_{Q_1}, Q_1, T_{Q_1}) * \cdots * (S_{Q_n}, Q_n, T_{Q_n}) \ s.t \ T_{Q_i} = S_{Q_{i+1}}$$

Interval Delay words, untimed of Timed Ipomsets

- Timed Ipomsets and Interval delay words
  - ▷ Timed Ipomsets : (P, σ_P, d_P).
  - ▷ Steps sequence (HDA)

$$(S_{Q_0}, Q_0, T_{Q_0}) * (S_{Q_1}, Q_1, T_{Q_1}) * \cdots * (S_{Q_n}, Q_n, T_{Q_n}) \ s.t \ T_{Q_i} = S_{Q_{i+1}}$$

  - ▷ Interval delay words : steps sequence interspersed with delays (start/termination of events).

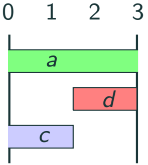
Interval Delay words, untimed of Timed Ipomsets

- Timed Ipomsets and Interval delay words
  - ▷ Timed Ipomsets : (P, σ_P, d_P).
  - ▷ Steps sequence (HDA)

$$(S_{Q_0}, Q_0, T_{Q_0}) * (S_{Q_1}, Q_1, T_{Q_1}) * \cdots * (S_{Q_n}, Q_n, T_{Q_n}) \ s.t \ T_{Q_i} = S_{Q_{i+1}}$$

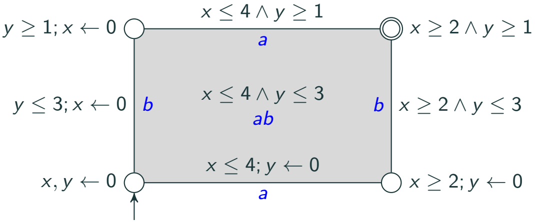
  - ▷ Interval delay words : steps sequence interspersed with delays (start/termination of events).

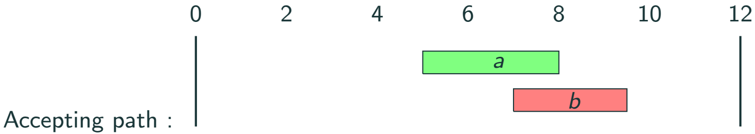
- Untimed of Timed Ipomsets
  - ▷ Untimed $unt((P, \sigma_P, d_P)) = (P, <_P, \dashrightarrow_P, S, T, \lambda)$

Ex : T = 

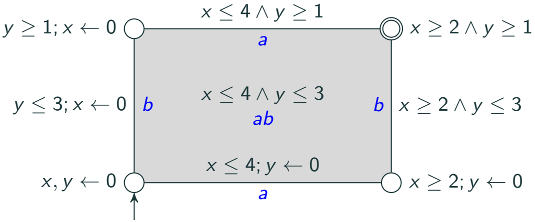
, $unt(T) = \begin{bmatrix} \bullet a \bullet \\ \bullet c \to d \end{bmatrix}$

- Example of accepting path

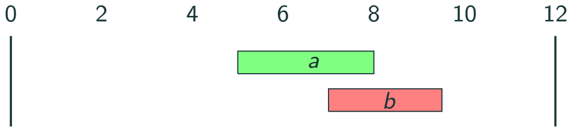




- Example of accepting path



Accepting path :



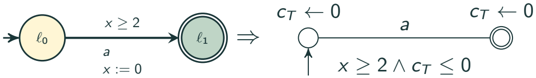
- The language of an HDTA A is :

$$L(A) = \{ev(\alpha) | \alpha \text{ accepting path in } X\}$$

- Contribution : **Embedding of TA into HDTA**

Let $\mathcal{A}$ be a timed automaton, we can transform it to express it as an HDTA, providing :

- ▷ **forcing immediate transition** : add a global clock x, for any transition
- ▷ Examples : TA(left), HDTA (right)



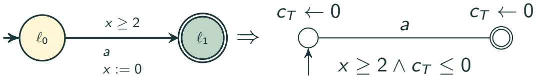
- Contribution : **Corollary**

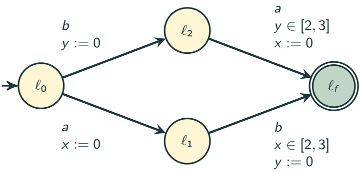
Language inclusion of HDTA is **undecidable**.

- Contribution : **Embedding of TA into HDTA**

Let $\mathcal{A}$ be a timed automaton, we can transform it to express it as an HDTA, providing :

- ▷ **forcing immediate transition** : add a global clock x, for any transition
- ▷ Examples : TA(left), HDTA (right)





$\Rightarrow$? (quizz if time)

- Contribution : **Corollary**

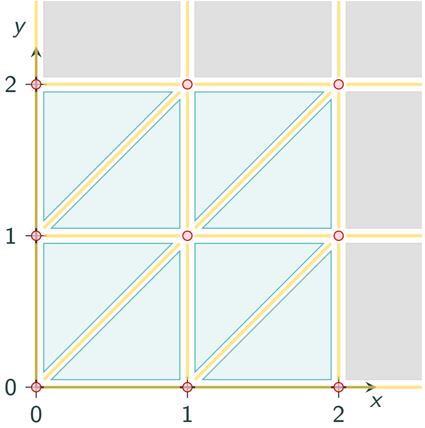
Language inclusion of HDTA is **undecidable**.

Untimed language inclusion is decidable

- Contribution : **Express Region Automata for HDTA**

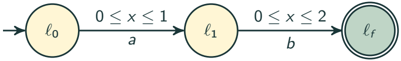
Untimed language inclusion is decidable

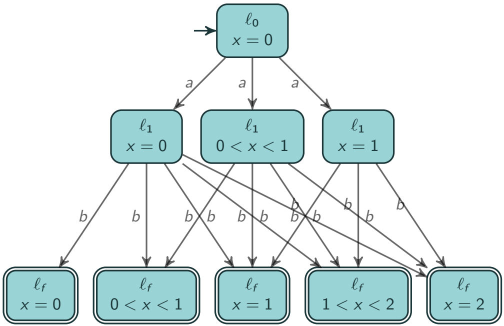
- Ex : Region of the constraint $0 \leq x, y \leq 2$



Region Automaton : example for one-clock TA

- A timed Automaton and its region automaton





- Reachability problem for TA

PSPACE (Alur et al, 1994) : correspondance between runs of TA and the one of the corresponding region automata.

Untimed language inclusion is decidable

- **Region equivalence**

Let $A = (\Sigma, C, L, \bot_L, \top_L, inv, exit)$ be an HDTA

▷ $M :=$ the maximal constant appearing in *inv*

▷ $\cong$: region equivalence on $\mathbb{R}^C_{\geq 0}$ defined as follows : for any $v, v' : C \to \mathbb{R}_{\geq 0}$:

- $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$ or $v(x), v'(x) > M$, $\forall x \in C$,
- $\{v(x)\} = 0 \Leftrightarrow \{v'(x)\} = 0, \forall x \in C$
- $\{v(x)\} \leq \{v(y)\} \Leftrightarrow \{v'(x)\} \leq \{v'(y)\}$

- Contribution : **Express untimed language**

For any HDTA A, $(unt(L(A))) = R(A)$.

- Consequence : **Untimed language inclusion is decidable**

Conclusion

Current & Future work

iiPomset & Timed iiPomset

- Expressivity of logics (LTL-like, FO) over iiPomset (Enzo Erlich PhD)
- Distance between Timed iiPomset

Robustness for HDTA

- Guard enlargment
- Delay perturbation
- Topological point of view

Appendix

- Delay words

Let us take a run $\pi = (\ell_0, v_0) \cdots \rightarrow \cdots (\ell_i, v_i) \cdots \rightarrow \cdots (\ell_n, v_n)$

▷ Delay move : $\delta : (\ell, v) \xrightarrow{d} (\ell, v + d)$
Label of delay move : $ev(\delta) = d$

▷ Action move : $\delta : (\ell, v) \xrightarrow{a_1} (\ell_1, v[y := 0])$
Label of action move : $ev(\delta) = a$

▷ Label of a run π :

$$ev((\ell_0, v_0) \rightarrow (\ell_1, v_1)) \cdots ev((\ell_{n-1}, v_{n-1}) \rightarrow (\ell_n, v_n))$$

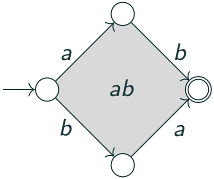
- Timed words

▷ Definition : $TW = \{w = (a_0, t_0) \cdots (a_n, t_n) t_{n+1} | \forall i = 0, \cdots n, t_i \leq t_{i+1}\} \subseteq (\Sigma \times \mathbb{R}_{\geq 0})^* \mathbb{R}_{\geq 0}$

▷ Concatenation : let $w = (a_0, t_0) \cdots (a_n, t_n) t_{n+1} w' = (a'_0, t'_0) \cdots (a'_n, t'_n) t'_{n+1} \in TW$ then :

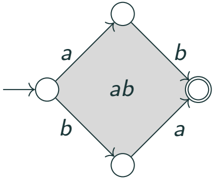
$$\boxed{ww' := (a_0, t_0) \cdots (a_n, t_n)(a'_0, t'_0) \cdots (a'_n, t'_n)(t_{n+1} + t'_{n+1}) \in TW}$$

Finally : $\mathcal{L}(A)$: the set of delay words labeling accepting path in the transition system.



- Higher Dimensional Automata A :
  - ▷ A tuple $(X, X_\perp, X_\top)$ where X is a finite **precubical set** and $X_\perp$ (*resp.* $X_\top$) $\subseteq X$ a **start (*resp.* accept) cell**.
  - ▷ Ex : start cell $X_\perp$: $\longrightarrow\bigcirc$, accept cell $X_\top$: $\circledcirc$
    X : $\{\longrightarrow\bigcirc$, $\circledcirc$, $\bigcirc$, ab $\} \cup \left\{ \xrightarrow{\lambda} | \lambda \in \{a, b\} \right\}$

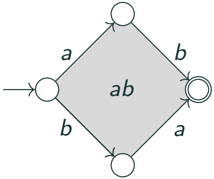
Definition of Higher Dimensional Automata



- Higher Dimensional Automata A :
  - ▷ A tuple $(X, X_\bot, X_\top)$ where X is a finite **precubical set** and $X_\bot$ (*resp.* $X_\top$) $\subseteq X$ a **start (*resp.* accept) cell**.
  - ▷ Ex : start cell $X_\bot$: $\longrightarrow$○ , accept cell $X_\top$: ◎

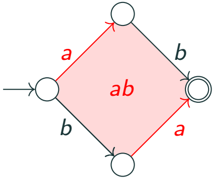
    X : { $\longrightarrow$○ , ◎, ○ , ab } $\cup \left\{ \xrightarrow{\lambda} | \lambda \in \{a, b\} \right\}$
- List of events
  - ▷ A **conclist** (concurrent list) : a finite, totally ordered ($\dashrightarrow$) $\Sigma-$labelled set.
  - ▷ Ex : $\{a, b\}$

Definition of Higher Dimensional Automata

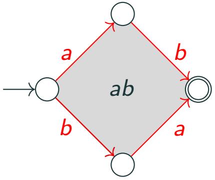


- Precubical set X :
  - ▷ A set of cells $\boldsymbol{X}$.
  - ▷ **List of active events** of a cell $x \in X$: a conclist $ev(x)$.
    Ex : $\{a\}$**, or** $\{b\}$ **or** $\{a, b\}$**.**

Definition of Higher Dimensional Automata



- Precubical set X :
  - ▷ A set of cells $\boldsymbol{X}$.
  - ▷ **List of active events** of a cell $x \in X$: a conclist $ev(x)$.
    Ex : $\{a\}$**, or** $\{b\}$ **or** $\{a, b\}$**.**
  - ▷ **The cells of a list of events** U : $\boldsymbol{X[U] = \{x \in X | ev(x) = U\}}$.
    Ex : $\boldsymbol{X[a]}$



- Precubical set X :
  - ▷ A set of cells $\boldsymbol{X}$.
  - ▷ **List of active events** of a cell $x \in X$: a conclist $ev(x)$.
    Ex : $\{a\}$, **or** $\{b\}$ **or** $\{a, b\}$.
  - ▷ **The cells of a list of events** U : $X[U] = \{x \in X | ev(x) = U\}$.
    Ex : $X[a]$
  - ▷ **Lower & Upper faces** : Let U and $A \subseteq U$ be conclists.
    $\delta_A^0 \backslash \delta_A^1$ represent **unstarting\terminating events** A :

$$\delta_A^0 : X[U] \to X[U - A], \delta_A^1 : X[U] \to X[U - A]$$

Path of HDA

- Paths in an HDA

Sequence $p = (x_0, \varphi_1, x_1, \cdots, x_{n-1}, \varphi_n, x_n)$ *s.t.*

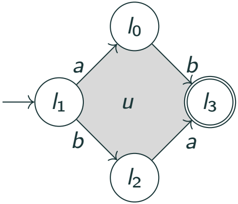
  - $x_i \in X$, where x_0 : start cell, x_n : accept cells
  - φ : face map type.
  - $ev(p_1 * p_2 * \cdots * p_n) = ev(p_1) * \cdots * ev(p_n)$

- Paths in an HDA

Sequence $p = (x_0, \varphi_1, x_1, \cdots, x_{n-1}, \varphi_n, x_n)$ *s.t.*

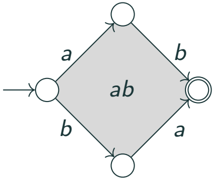
- ▷ $x_i \in X$, where x_0 : start cell, x_n : accept cells
- ▷ φ : face map type.
- ▷ $ev(p_1 * p_2 * \cdots * p_n) = ev(p_1) * \cdots * ev(p_n)$

- Example of a 2−events HDA



Example of an accepting path :

$$\alpha_0 = l_0 \nearrow^{ab} u \searrow_{ab} l_3,\ ev(\alpha_0) = \left(\begin{bmatrix} a\bullet \\ b\bullet \end{bmatrix} * \begin{bmatrix} \bullet a \\ \bullet b \end{bmatrix} \right)$$



- Precubical set
  - ▷ Sets $(X_n)_n$
  - ▷ A set of functions $\left(\delta^{\varepsilon}_{i,n} : X_n \mapsto X_{n-1}\right)_{n>0, i\in\{1,\cdots,n\}, \varepsilon\in\{0,1\}}$ such that :

$$\boxed{\delta^{\varepsilon'}_{j,n} \circ \delta^{\varepsilon}_{i,n+1} = \delta^{\varepsilon}_{i-1,n} \circ \delta^{\varepsilon}_{j,n+1}, \forall i, j}$$

.

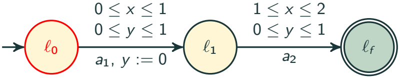
- Application in HDA

A precubical set on a finite alphabet Σ :

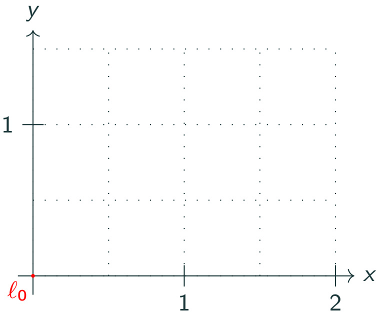
$$X = \left(X, ev, \{\delta^0_{A,U}, \delta^1_{A,U} | U \in C, A \subseteq U\}\right)$$

where C is the set of conclist over Σ

- Timed automaton $\mathcal{A}$:

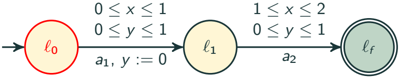


- Run with delay perturbations of at most $\delta = 0.2$

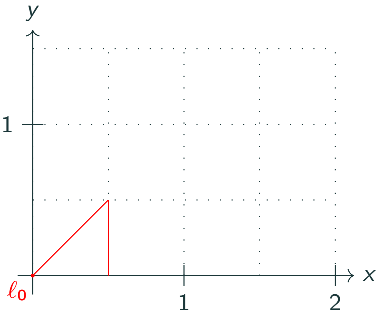


Future work : What about the robustness ?

- Timed automaton $\mathcal{A}$:

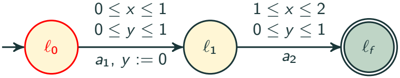


- Run with delay perturbations of at most $\delta = 0.2$

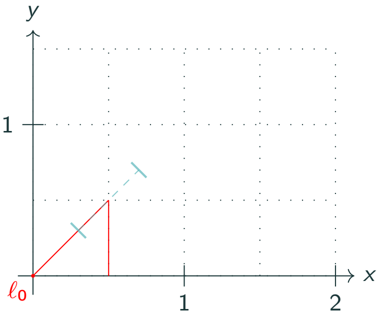


Future work : What about the robustness ?

- Timed automaton $\mathcal{A}$:

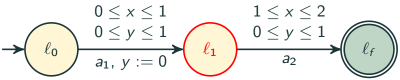


- Run with delay perturbations of at most $\delta = 0.2$

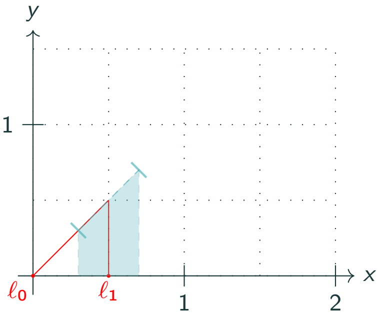


Future work : What about the robustness ?

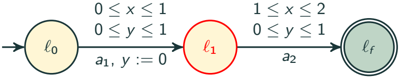
- Timed automaton $\mathcal{A}$:



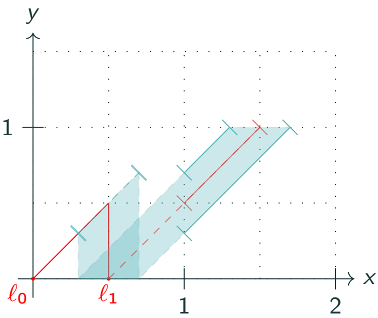
- Run with delay perturbations of at most $\delta = 0.2$



- Timed automaton $\mathcal{A}$:

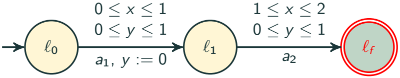


- Run with delay perturbations of at most $\delta = 0.2$

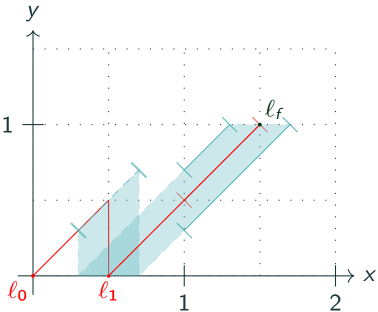


Future work : What about the robustness ?

- Timed automaton $\mathcal{A}$:

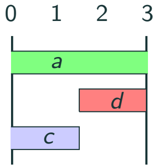


- Run with delay perturbations of at most $\delta = 0.2$



Future work : robustness on HDTA

- No timing perturbation : c and d are not in concurrency



- timing perturbation. Let us introduce a 0.1 delay on the end date of c :