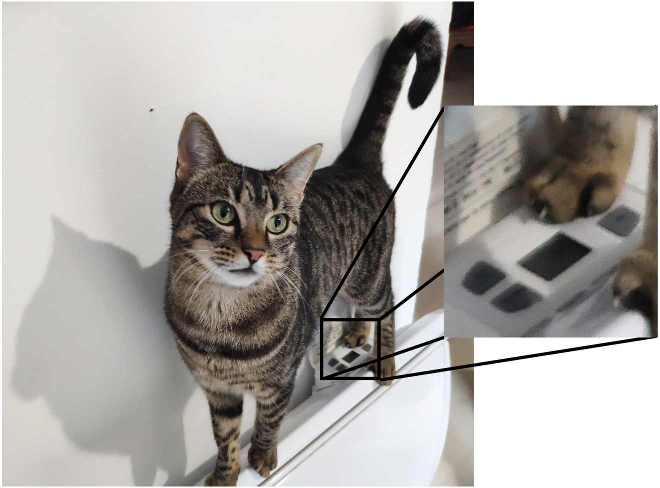
PhD Defence
Soutenance de Doctorat
Automata on Timed Structures

Samy JAZIRI
Supervised by *Patricia Bouyer-Decitre* and *Nicolas Markey*

September 24, 2019

Heater Monitoring

Heater Monitoring



Heater Monitoring

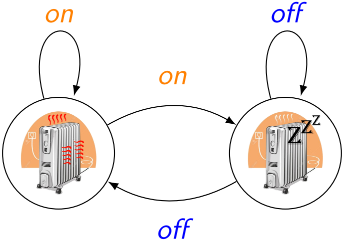
Heater Monitoring

States of the Heater

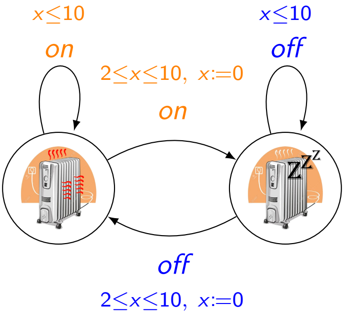
heating

doing nothing

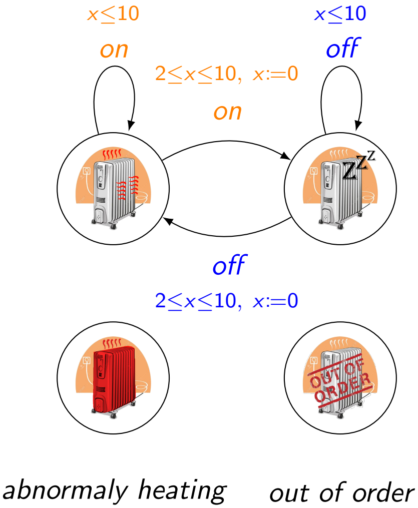
Modeling



Modeling

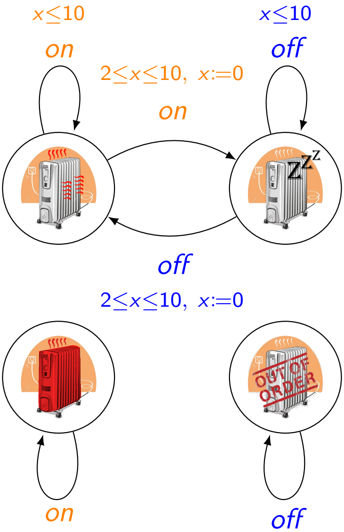


Modeling



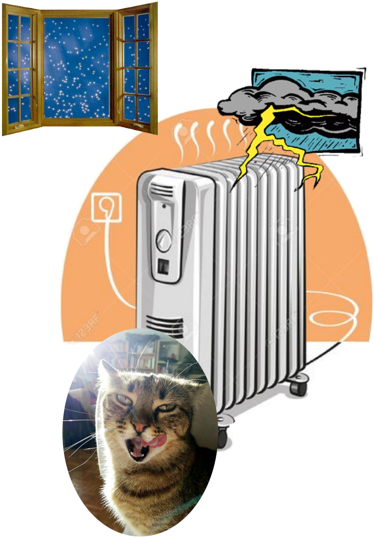
x≤10
on
x≤10
off
2≤x≤10, x:=0
on
off
2≤x≤10, x:=0
ZZZ
OUT OF ORDER
abnormaly heating
out of order

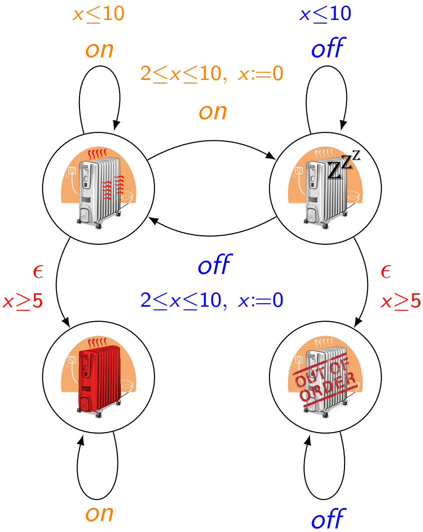
Modeling



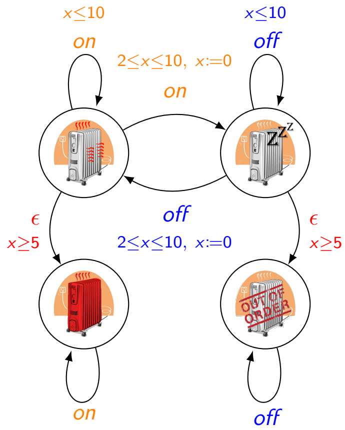
$x \leq 10$
on
$x \leq 10$
off
$2 \leq x \leq 10,\ x := 0$
on
off
$2 \leq x \leq 10,\ x := 0$
on
off

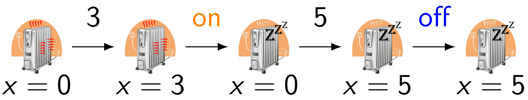
Modeling



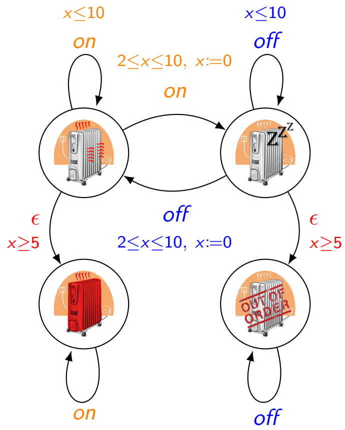


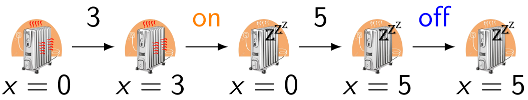
Executions in the Model





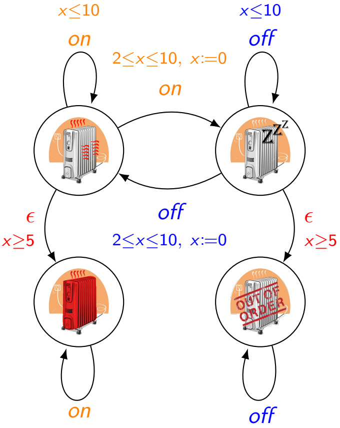
Executions in the Model

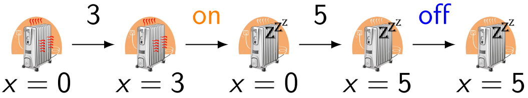




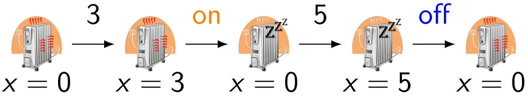
3·on·5·off

Executions in the Model

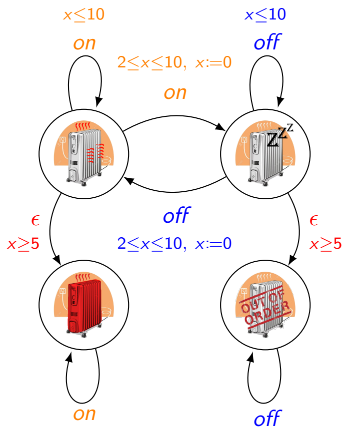


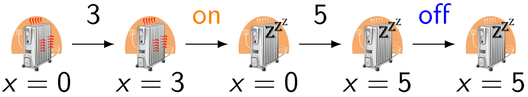


$3 \cdot \text{on} \cdot 5 \cdot \text{off}$

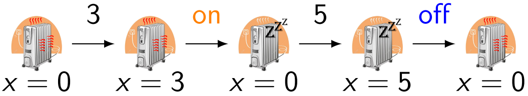


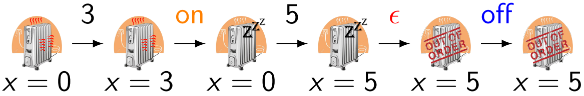
Executions in the Model



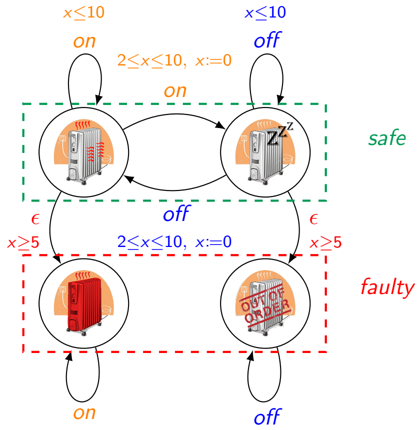


3·on·5·off

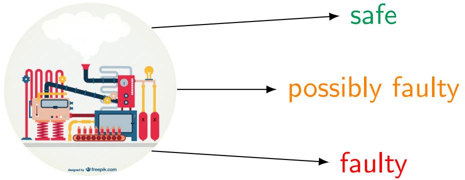




Diagnosis

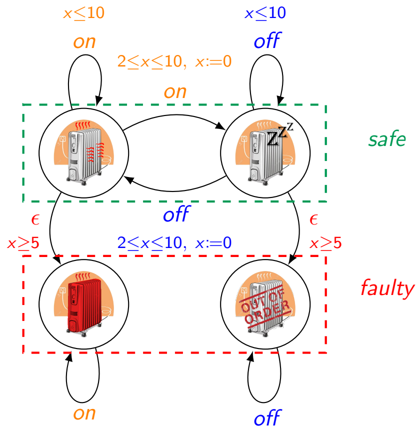


$x \leq 10$
on
$x \leq 10$
off
$2 \leq x \leq 10,\ x := 0$
on
safe
ϵ
$x \geq 5$
off
$2 \leq x \leq 10,\ x := 0$
ϵ
$x \geq 5$
faulty
on
off

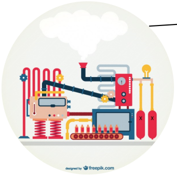


safe
possibly faulty
faulty

Diagnosis

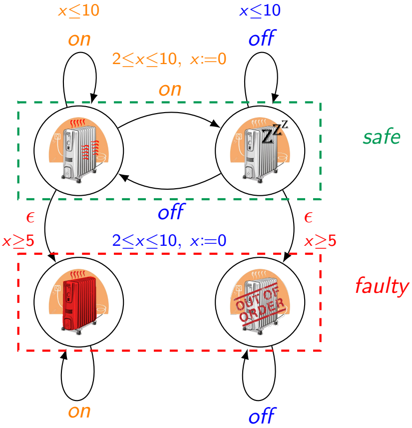


$1 \cdot \text{on} \cdot 3 \cdot \text{on}$

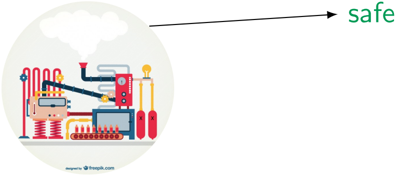


safe

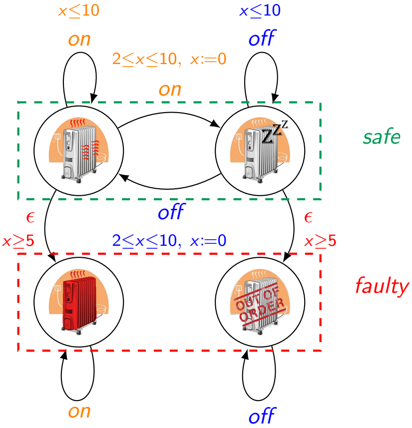
Diagnosis



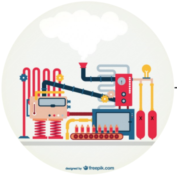
1·on·3·on



Diagnosis

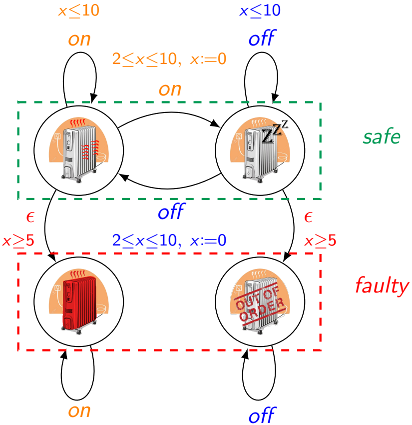


$1 \cdot \text{on} \cdot 3 \cdot \text{on} \cdot 8$

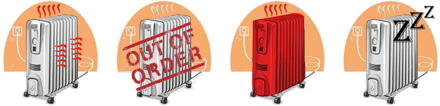


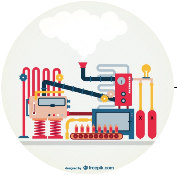
possibly faulty

Diagnosis



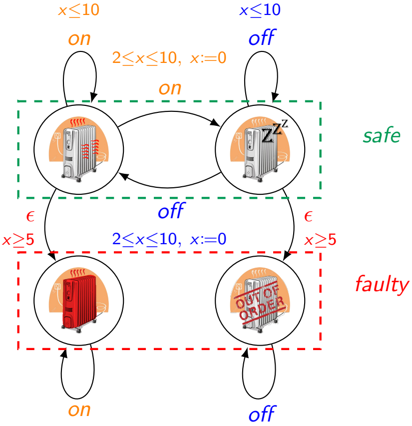
$1 \cdot \text{on} \cdot 3 \cdot \text{on} \cdot 8$



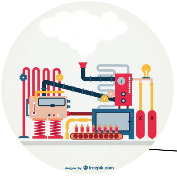


→ possibly faulty

Diagnosis

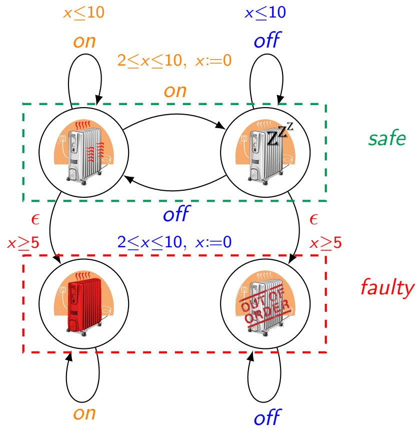


$1 \cdot \mathit{on} \cdot 3 \cdot \mathit{on} \cdot 8 \cdot \mathit{on}$

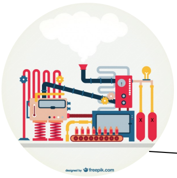


faulty

Diagnosis



$1 \cdot on \cdot 3 \cdot on \cdot 8 \cdot on$



faulty

Weakness of Times Automata

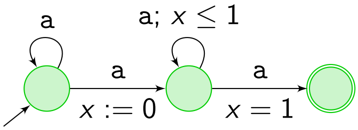


Figure: A non-determinizable timed automaton [AD94]

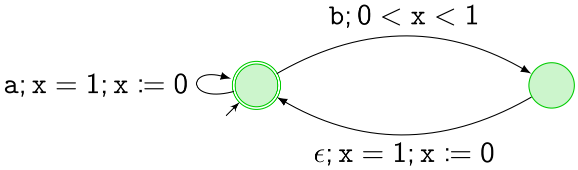


Figure: Timed automaton where ϵ-transitions can't be removed [BDGP98]

My Work

- **Step 1 :** Determinization - General Powerset Construction
  - *Definition of an adapted model - Automata on Timed Structures*
  - *Definition of a general powerset construction for Automata on Timed Structures*
  - *Relations between the new powerset construction and previous determinization results*

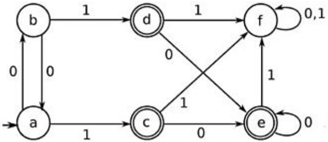
My Work

- **Step 1 :** Determinization - General Powerset Construction
  - *Definition of an adapted model - Automata on Timed Structures*
  - *Definition of a general powerset construction for Automata on Timed Structures*
  - *Relations between the new powerset construction and previous determinization results*
- **Step 2 :** Diagnosis - A diagnoser for 1-clock timed automata
  - *Introduction of an adapted data structure - Regular Timed Interval*
  - *Powerset Construction + Regular Timed Interval = diagnoser*

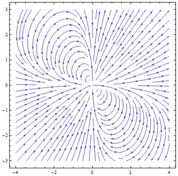
My Work

- **Step 1 :** Determinization - General Powerset Construction
  - *Definition of an adapted model - Automata on Timed Structures*
  - *Definition of a general powerset construction for Automata on Timed Structures*
  - *Relations between the new powerset construction and previous determinization results*
- **Step 2 :** Diagnosis - A diagnoser for 1-clock timed automata
  - *Introduction of an adapted data structure - Regular Timed Interval*
  - *Powerset Construction + Regular Timed Interval = diagnoser*
- **Step 3 :** Implementation - The power of precomputation
  - *Implementation of the diagnoser of Stavros Tripakis.*
  - *Implementation of our diagnoser with some precomputation.*
  - *Comparison of the performance.*

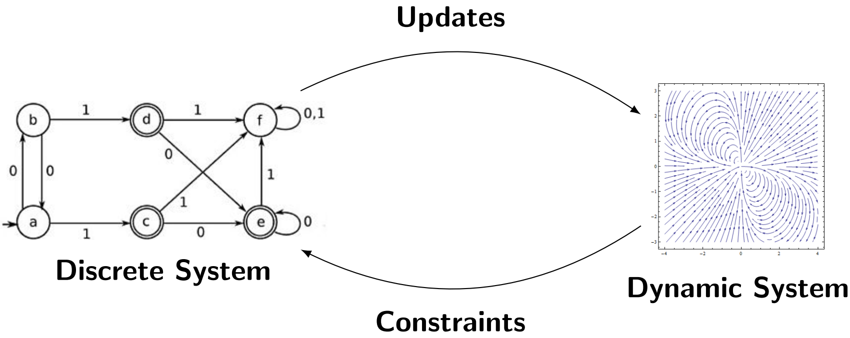
General Idea



Discrete System

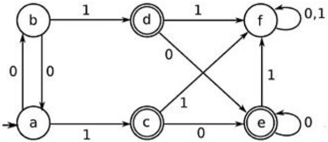


Dynamic System


Updates
b
d
f
a
c
e
1
1
0,1
0
0
0
1
1
1
1
0
0
Discrete System
Dynamic System
Constraints

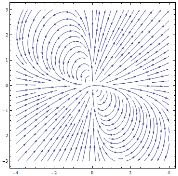
General Idea

Updates



Discrete System

Guards

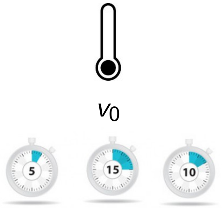


Dynamic System

Constraints

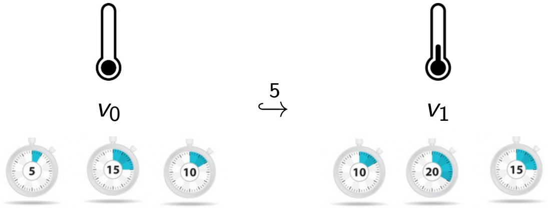
Timed Domains

Une horloge donne l'heure, nous sommes bien d'accord, elle passe même ses heures à ne faire que cela, mais elle ne montre rien de ce qu'est le temps en amont de ce processus d'actualisation. Elle dissimule plutôt le temps derrière le masque convaincant d'une mobilité parfaitement régulière.
– Etienne Klein, Les tactiques de Kronos



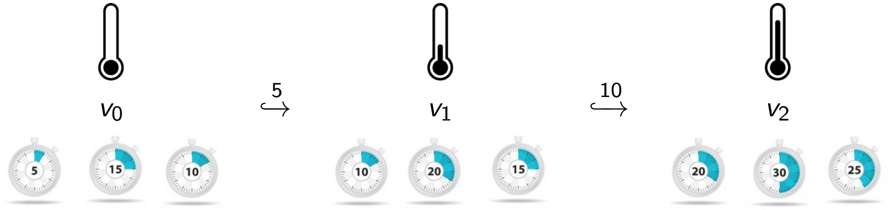
Timed Domains

Une horloge donne l'heure, nous sommes bien d'accord, elle passe même ses heures à ne faire que cela, mais elle ne montre rien de ce qu'est le temps en amont de ce processus d'actualisation. Elle dissimule plutôt le temps derrière le masque convainçant d'une mobilité parfaitement régulière.
– Etienne Klein, Les tactiques de Kronos



Timed Domains

> *Une horloge donne l'heure, nous sommes bien d'accord, elle passe même ses heures à ne faire que cela, mais elle ne montre rien de ce qu'est le temps en amont de ce processus d'actualisation. Elle dissimule plutôt le temps derrière le masque convaincant d'une mobilité parfaitement régulière.*
> *– Etienne Klein, Les tactiques de Kronos*



Timed Domains

Definition

A *timed domain* is a couple $\langle V, \hookrightarrow \rangle$ where V is a set called *value space* and $\hookrightarrow$ is a function from $\mathbb{R}_{\geq 0}$ to $\mathcal{P}(V \times V)$ called *delay transition*, such that

- for all $v \in V$, $(v, v) \in \hookrightarrow(0)$
- for all $v \in V$ and $d \in \mathbb{R}_{\geq 0}$, there exists $v' \in V$ such that $(v, v') \in \hookrightarrow(d)$
- for all $d, d' \in \mathbb{R}_{\geq 0}$, for all $v, v', v'' \in V$,

$$\text{if } (v, v') \in \hookrightarrow(d) \text{ and } (v', v'') \in \hookrightarrow(d') \text{ then } (v, v'') \in \hookrightarrow(d + d')$$

Timed Domains

Definition

A *timed domain* is a couple $\langle V, \hookrightarrow \rangle$ where V is a set called *value space* and $\hookrightarrow$ is a function from $\mathbb{R}_{\geq 0}$ to $\mathcal{P}(V \times V)$ called *delay transition*, such that

- for all $v \in V$, $(v, v) \in \hookrightarrow(0)$
- for all $v \in V$ and $d \in \mathbb{R}_{\geq 0}$, there exists $v' \in V$ such that $(v, v') \in \hookrightarrow(d)$
- for all $d, d' \in \mathbb{R}_{\geq 0}$, for all $v, v', v'' \in V$,

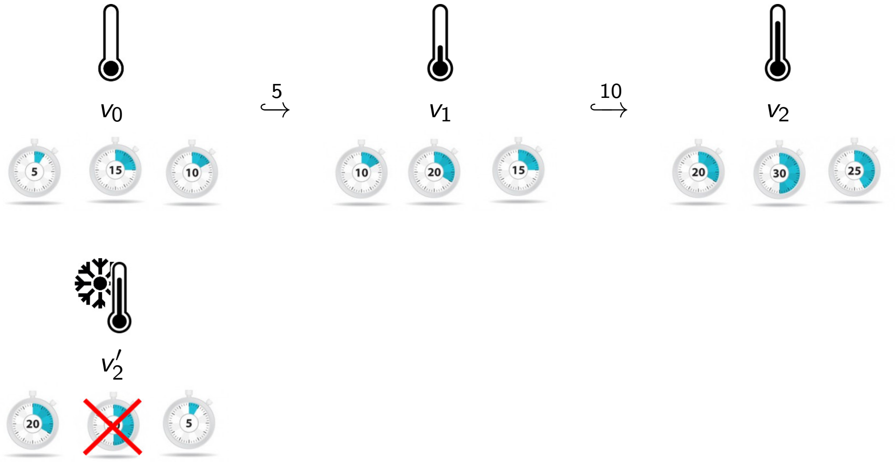
$$\text{if } (v, v') \in \hookrightarrow(d) \text{ and } (v', v'') \in \hookrightarrow(d') \text{ then } (v, v'') \in \hookrightarrow(d + d')$$

Example: The M-bounded clock domain is

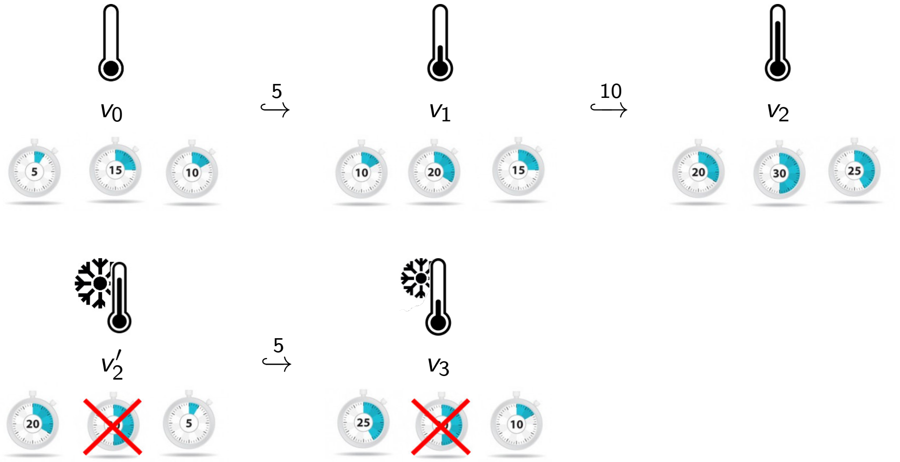
$$\langle [0; M] \cup \{\infty\}, \hookrightarrow \rangle$$

In the 5-bounded clock domain $3 \overset{1}{\hookrightarrow} 4 \overset{1.2}{\hookrightarrow} \infty$

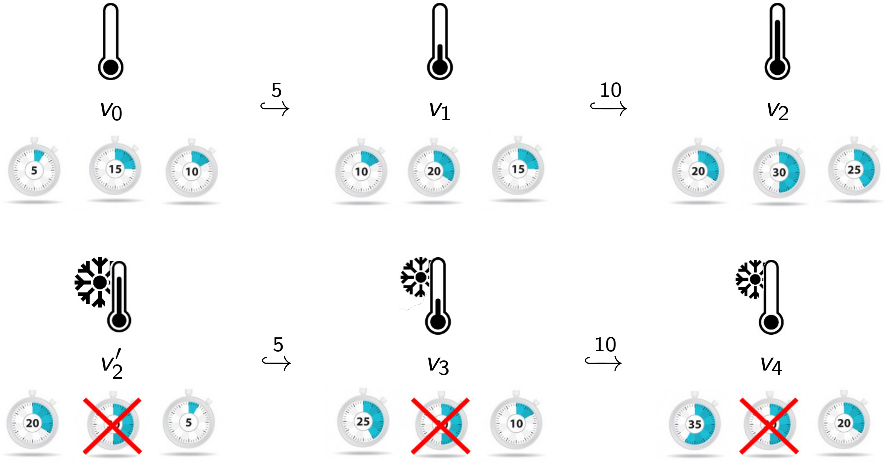
Timed Structures



Timed Structures



Timed Structures



Timed Structures

Definition

Let $D = \langle V, \hookrightarrow \rangle$ be a timed domain. An *update set* on D is a subset of V^V.

Definition

A *timed structure* is a triple $T = \langle V, \hookrightarrow, U \rangle$ where $\langle V, \hookrightarrow \rangle$ is a timed domain and U is an update set on $\langle V, \hookrightarrow \rangle$.

Timed Structures

Definition

Let $D = \langle V, \hookrightarrow \rangle$ be a timed domain. An *update set* on D is a subset of V^V.

Definition

A *timed structure* is a triple $T = \langle V, \hookrightarrow, U \rangle$ where $\langle V, \hookrightarrow \rangle$ is a timed domain and U is an update set on $\langle V, \hookrightarrow \rangle$.

Example: The one-dimensional clock structure can be equipped with the two updates **id** and **0** to form the M-clock structure :

$$\langle [0; M] \cup \{\infty\}, \hookrightarrow, \{\mathbf{id}, \mathbf{0}\} \rangle$$

Guards

Timed Structure (TS) $\langle [0;2] \cup \{\infty\}, \hookrightarrow, \{\mathbf{id}, \mathbf{0}\} \rangle$

Definition

Given a timed structure $\mathcal{S} = (V, \hookrightarrow, U)$ we call *guard* on $\mathcal{S}$ any subset of $V \times U$.

Guards

Timed Structure (TS)	$\langle [0;2] \cup \{\infty\}, \hookrightarrow, \{\mathbf{id}, \mathbf{0}\}\rangle$
Guard	$[0,1[\times \{\mathbf{0}\} \cup \{(\frac{\pi}{2}, \mathbf{id})\}$
TA Notations	$0 \leq x < 1; x := 0$ **or** $x = \frac{\pi}{2}$

Definition

Given a timed structure $\mathcal{S} = (V, \hookrightarrow, U)$ we call *guard* on $\mathcal{S}$ any subset of $V \times U$.

Guards

Timed Structure (TS)	$\langle [0;2] \cup \{\infty\}, \hookrightarrow, \{\mathbf{id}, \mathbf{0}\} \rangle$
Guard	$[0,1[\times \{\mathbf{0}\} \cup \{(\frac{\pi}{2}, \mathbf{id})\}$
TA Notations	$0 \leq x < 1; x := 0$ **or** $x = \frac{\pi}{2}$
Guard Basis	$\mathcal{G} = \{\{0\},]0;1[, \{1\},]1;2[, \{2\}, \{\infty\}\} \times \{\mathbf{0}, \mathbf{id}\}$
TA Notations	$x = 0, 0 < x < 1, x = 1, 1 < x < 2, x = 2, x > 2$ with a potential reset on x
Example	$[1,2[\times \{\mathbf{0}, \mathbf{id}\} =$ $\{(1,\mathbf{0})\} \cup]1,2[\times \{\mathbf{0}\} \cup \{(1,\mathbf{id})\} \cup]1,2[\times \{\mathbf{id}\}$

Definition

Given a timed structure $\mathcal{S} = (V, \hookrightarrow, U)$ we call *guard* on $\mathcal{S}$ any subset of $V \times U$.

Guards

Timed Structure (TS)	$\langle [0;2] \cup \{\infty\}, \hookrightarrow, \{\mathbf{id}, \mathbf{0}\} \rangle$
Guard	$[0,1[\times \{\mathbf{0}\} \cup \{(\frac{\pi}{2}, \mathbf{id})\}$
TA Notations	$0 \leq x < 1; x := 0$ **or** $x = \frac{\pi}{2}$
Guard Basis	$\mathcal{G} = \{\{0\},]0;1[, \{1\},]1;2[, \{2\}, \{\infty\}\} \times \{\mathbf{0}, \mathbf{id}\}$
TA Notations	$x = 0, 0 < x < 1, x = 1, 1 < x < 2, x = 2, x > 2$ with a potential reset on x
Example	$[1,2[\times \{\mathbf{0}, \mathbf{id}\} =$ $\{(1,\mathbf{0})\} \cup]1,2[\times \{\mathbf{0}\} \cup \{(1,\mathbf{id})\} \cup]1,2[\times \{\mathbf{id}\}$
Guarded TS	$\langle [0;2] \cup \{\infty\}, \hookrightarrow, \{\mathbf{id}, \mathbf{0}\}, \mathcal{G} \rangle$

Definition

Let $\mathcal{S} = \langle V, \hookrightarrow, U \rangle$ be a timed structure and let G be a guard basis on $\mathcal{S}$. The tuple $\mathcal{S}_G = \langle V, \hookrightarrow, U, G \rangle$, is named *guarded timed structure*.

Automata on Timed Structures

Definition

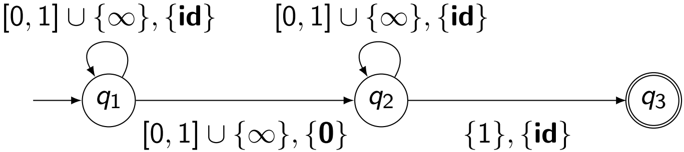
Fix $\mathcal{S} = \langle V, \hookrightarrow, U \rangle$, a timed structure. An automaton on $\mathcal{S}$ is a tuple

$$A = \langle Q, I, T, F \rangle$$

where Q is a finite set of states, $I \subseteq Q \times V$ is the set of initial configurations, $T \subseteq Q \times V \times U \times Q$ is the transition relation, and $F \subseteq Q$ is the set of final states.
We write $\mathbb{A}(\mathcal{S})$ for the set of automata over the timed structure $\mathcal{S}$.

Automata on Timed Structures

If $\mathcal{S}_{\mathcal{G}}$ is a guarded timed structure and the guards of an automaton A are decomposable on the guard basis $\mathcal{G}$ we say that A is compatible with $\mathcal{G}$ and write $A \in \mathbb{A}(\mathcal{S}_{\mathcal{G}})$.



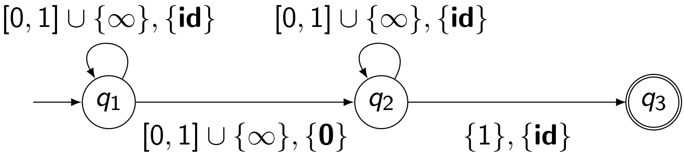
An automaton $A = \langle Q, I, T, F \rangle$ induces an labeled infinite-state transition system on $Q \times V$.

$$\begin{aligned}
(q, v) \xrightarrow{d} (q', v') \quad &\Leftrightarrow \quad q = q' \text{ and } v \overset{d}{\hookrightarrow} v' \\
(q, v) \xrightarrow{u} (q', v') \quad &\Leftrightarrow \quad (q, v, u, q') \in T \text{ and } v' = u(v).
\end{aligned}$$

Example: A trace of an automaton in $\mathbb{A}(\langle [0;2] \cup \{\infty\}, \hookrightarrow, \{\mathbf{id}, \mathbf{0}\}\rangle)$.

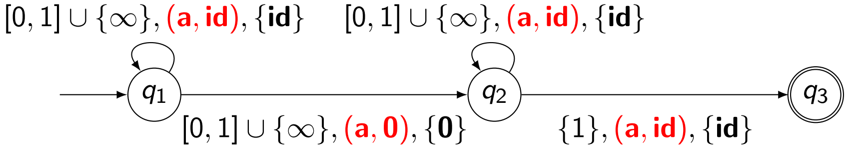
$$(q_1, 0) \xrightarrow{1.2} (q_1, 1.2) \xrightarrow{\mathbf{0}} (q_2, 0) \xrightarrow{3} (q_2, \infty) \xrightarrow{\mathbf{id}} (q_3, \infty)$$

Controls



$[0,1] \cup \{\infty\}, \{\mathbf{id}\}$
$[0,1] \cup \{\infty\}, \{\mathbf{id}\}$
q_1
q_2
q_3
$[0,1] \cup \{\infty\}, \{\mathbf{0}\}$
$\{1\}, \{\mathbf{id}\}$

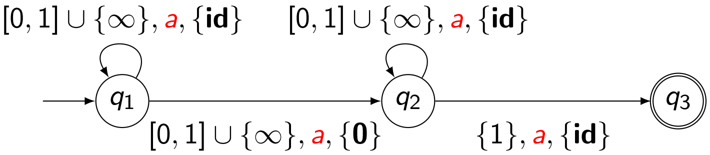
Full Controls



All runs recognizing $(a, \mathbf{id}) \cdot (a, \mathbf{0}) \cdot 1 \cdot (a, \mathbf{id})$:

$$(q_1, 0) \xrightarrow{(a,\mathbf{id})} (q_1, 0) \xrightarrow{(a,\mathbf{0})} (q_2, 0) \xrightarrow{1} (q_2, 1) \xrightarrow{(a,\mathbf{id})} \begin{cases} (q_3, 1) \\ (q_2, 1) \end{cases}$$

Timed Controls



All runs recognizing $a \cdot a \cdot 1 \cdot a$:

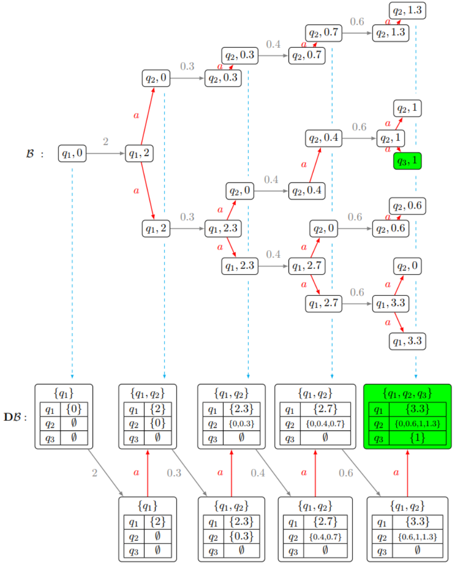
$$(q_1,0) \xrightarrow{a} \begin{cases} (q_1,0) \xrightarrow{a} \begin{cases} (q_1,0) \xrightarrow{1} (q_1,1) \xrightarrow{a} \begin{cases} (q_1,1) \\ (q_2,0) \end{cases} \\ (q_2,0) \xrightarrow{1} (q_2,1) \xrightarrow{a} \begin{cases} (q_2,1) \\ (q_3,1) \end{cases} \end{cases} \\ (q_2,0) \xrightarrow{a} (q_2,0) \xrightarrow{1} (q_2,1) \xrightarrow{a} \begin{cases} (q_2,1) \\ (q_3,1) \end{cases} \end{cases}$$

General Powerset Construction

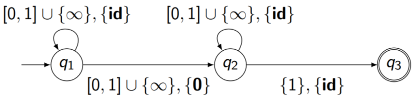
Theorem

Let Σ a finite alphabet, $\mathcal{S}$ a timed structure and A an automaton on $\mathcal{S}$ equipped with a timed control κ over Σ and without silent transitions. A is Σ-determinizable ; i.e. there exists A' an automaton on $\mathbf{D}_{A,\kappa}\mathcal{S}$ equipped with a timed control $\mathbf{D}\kappa$ such that A' is deterministic w.r.t $\mathbf{D}\kappa$ and A' recognize the same language as A.
Moreover, if A is finitely representable, we can construct a determinized automaton of A with a finite representation.

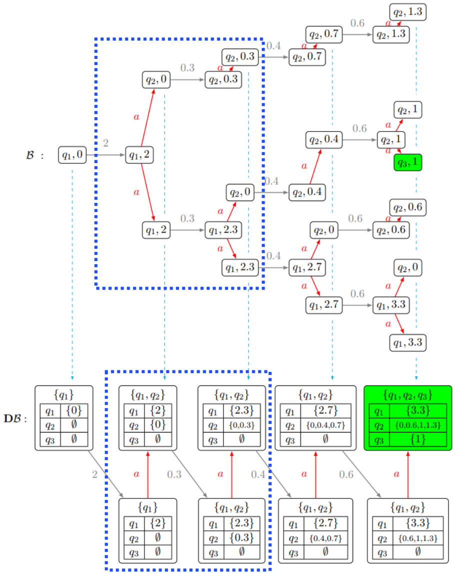
General Powerset Construction



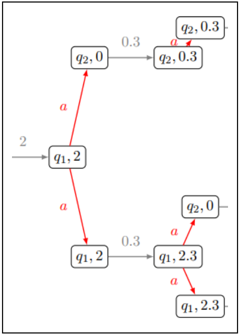
Simulation of $a \cdot 2 \cdot a \cdot 0.3 \cdot a \cdot 0.4 \cdot a \cdot 0.6 \cdot a$

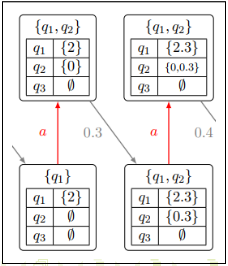


General Powerset Construction

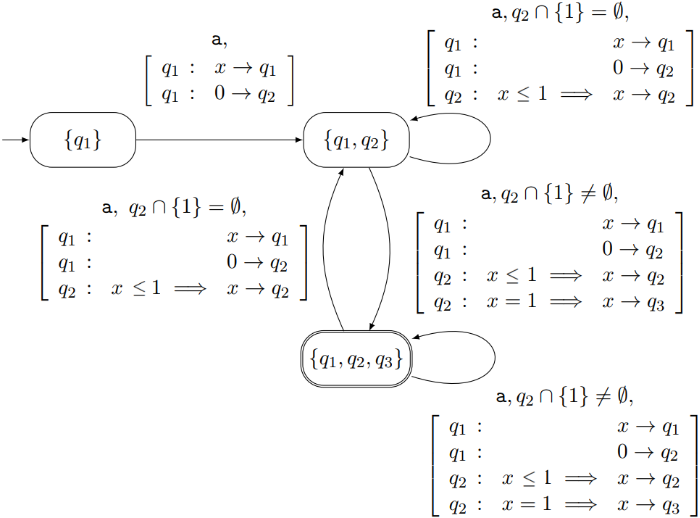


Simulation of $a \cdot 2 \cdot a \cdot 0.3 \cdot a \cdot 0.4 \cdot a \cdot 0.6 \cdot a$





Example of Powerset Construction



Classical Determinization

- Bounded Sets Case $\longrightarrow$ [BBBB09] [1]
- Strongly non-Zeno Timed Automata
- 0-Bounded Timed Automata [OW04] [2]
- Integer Reset Timed Automata [SPKM08] [3]
- Finally Imprecise Timed Automata

[1] **Baier, Bertrand, Bouyer, and Brihaye.** When Are Timed Automata Determinizable? *ICALP09*

[2] **Ouaknine and Worrell.** On the language inclusion problem for timed automata: Closing a decidability gap. *LICS04*

[3] **Suman, Pandya, Krishna and Manasa.** Timed Automata with Integer Resets: Language Inclusion... *FORMATS08*

Input-determined models

Theorem

Let Σ a finite alphabet, $\mathcal{S}_G$ a guarded deterministic timed structure where U is finite and $A \in \mathbb{A}(\mathcal{S}_G)$ equipped with a compatible Σ-full control. Then A can be Σ-determinized in the same guarded timed structure $\mathcal{S}_G$.

[4] **Alur, Fix, and Henzinger**. A determinizable class of timed automata. *CAV94*

[5] **Abdulla, Atig, and Stenman**. Dense-timed pushdown automata *LICS12*

[6] **Bhave and Guha**. Adding dense-timed stack to integer reset timed automata. *RP17*

Input-determined models

Theorem

Let Σ a finite alphabet, $\mathcal{S}_G$ a guarded deterministic timed structure where U is finite and $A \in \mathbb{A}(\mathcal{S}_G)$ equipped with a compatible Σ-full control. Then A can be Σ-determinized in the same guarded timed structure $\mathcal{S}_G$.

Applications.

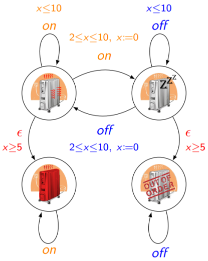
- Event-clock automata [AFH94] [4]
- (Timed) Visibly Pushdown Automata [AAS] [5]
- Strict Integer Reset Timed Automata [BG17] [6]

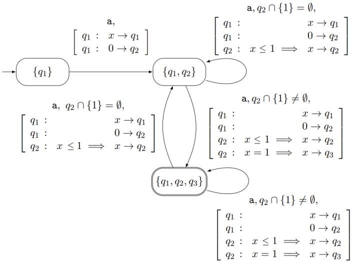
[4] **Alur, Fix, and Henzinger**. A determinizable class of timed automata. *CAV94*
[5] **Abdulla, Atig, and Stenman**. Dense-timed pushdown automata *LICS12*
[6] **Bhave and Guha**. Adding dense-timed stack to integer reset timed automata. *RP17*

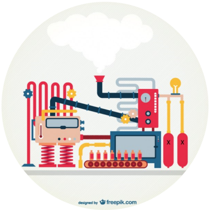
Step 2

- **Step 1 :** Determinization - General Powerset Construction
  - *Definition of an adapted model - Automata on Timed Structures*
  - *Definition of a general powerset construction for Automata on Timed Structures*
  - *Relations between the new powerset construction and previous determinization results*
- **Step 2 :** Diagnosis - A diagnoser for 1-clock timed automata
  - *Introduction of an adapted data structure - Regular Timed Interval*
  - *Powerset Construction* $+$ *Regular Timed Interval* $=$ *diagnoser*
- **Step 3 :** Implementation - The power of precomputation
  - *Implementation of the diagnoser of Tripakis.*
  - *Implementation of our diagnoser with some precomputation.*
  - *Comparison of the performance.*

Toward Diagnosis







What's left to be done

Timed Automaton	Powerset construction
$\xrightarrow{a}$	$\mathbf{U}_a$
$\xrightarrow{d}$	$\mathbf{U}_d$

What's left to be done

Timed Automaton	Powerset construction	
$\xrightarrow{a}$	$\mathbf{U}_a$	easy to compute
$\xrightarrow{d}$	$\mathbf{U}_d$	

What's left to be done

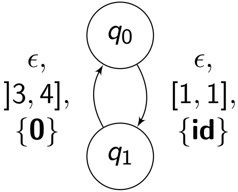
Timed Automaton	Powerset construction	
$\xrightarrow{a}$	$\mathbf{U}_a$	easy to compute
$\xrightarrow{d}$	$\mathbf{U}_d$	difficult to compute

What's left to be done

Timed Automaton	Powerset construction	
$\xrightarrow{a}$	$\mathbf{U}_a$	easy to compute
$\xrightarrow{d}$	$\mathbf{U}_d$	difficult to compute

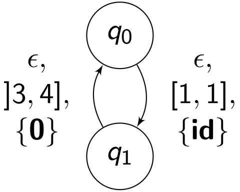
$$\begin{tikzcd}
\nu \arrow[r, "\mathbf{U}_d"] \arrow[d, dotted] & \nu' \arrow[d, dotted] \\
\mathbf{I} \arrow[r, "{\epsilon, +d}"'] & \mathbf{I}'
\end{tikzcd}$$

Regular Timed Interval



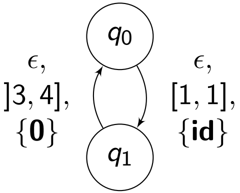
	$\mathbf{U}_0$
q_0 :	$\{0\}$
q_1 :	$\emptyset$

Regular Timed Interval



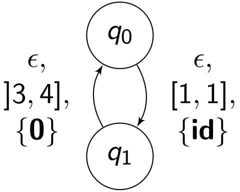
	$\mathbf{U}_0$	$\mathbf{U}_2$
q_0 :	$\{0\}$	$\{2\}$
q_1 :	$\emptyset$	$\{2\}$

Regular Timed Interval



	$\mathbf{U}_0$	$\mathbf{U}_2$	$\mathbf{U}_{3.5}$
q_0 :	$\{0\}$	$\{2\}$	$\{3.5\} \cup [0; 0.5[$
q_1 :	$\emptyset$	$\{2\}$	$\{3.5\}$

Regular Timed Interval



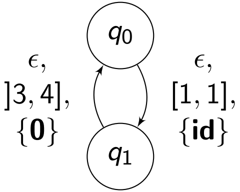
	$\mathbf{U}_0$	$\mathbf{U}_2$	$\mathbf{U}_{3.5}$	$\mathbf{U}_8$
q_0 :	$\{0\}$	$\{2\}$	$\{3.5\} \cup [0; 0.5[$	$\{8\} \cup [4,5[\cup [0,2[$
q_1 :	$\emptyset$	$\{2\}$	$\{3.5\}$	$\{8\} \cup [4,5[\cup [1,2[$

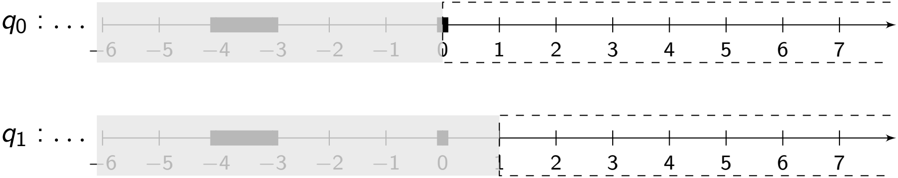
Regular Timed Interval

$\mathbf{U}_0$

q_0 : $\{0\}$

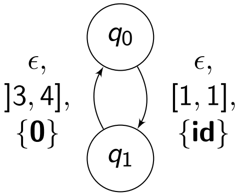
q_1 : $\emptyset$

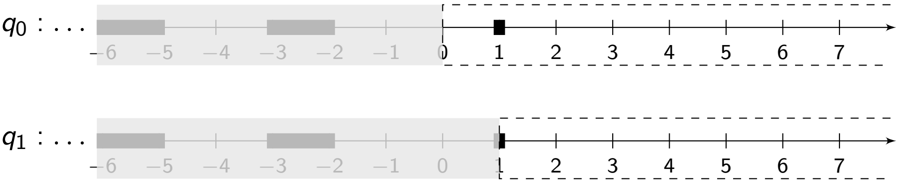




Regular Timed Interval

	$\mathbf{U}_1$
q_0 :	$\{1\}$
q_1 :	$\{1\}$

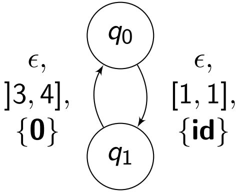


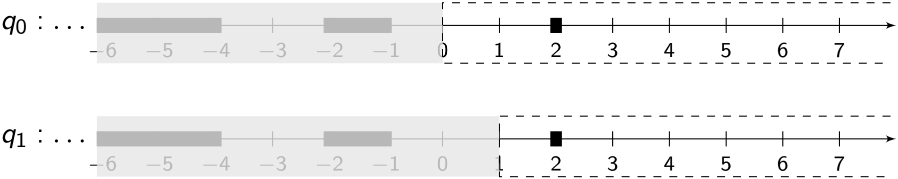


$\mathbf{U}_2$

q_0 : $\{2\}$

q_1 : $\{2\}$



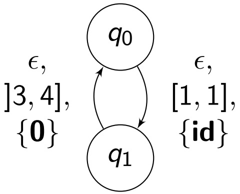


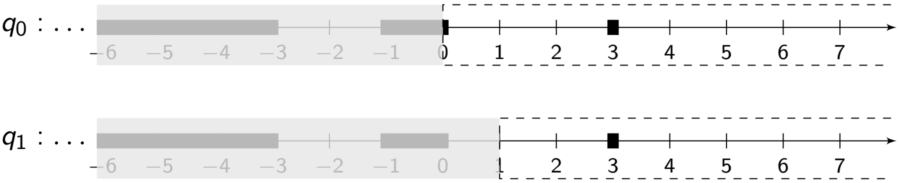
Regular Timed Interval

$\mathbf{U}_3$

q_0 : $\{3\}$

q_1 : $\{3\}$



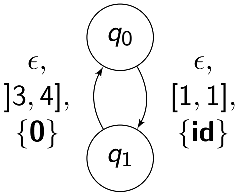


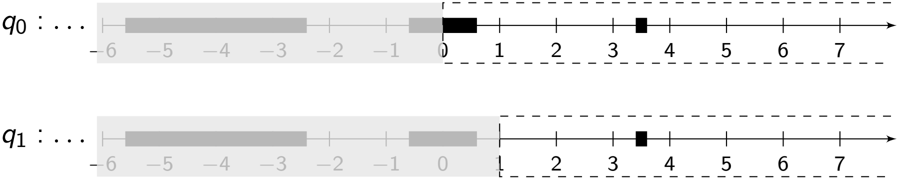
Regular Timed Interval

$\mathbf{U}_{3.5}$

q_0 : $\{3.5\} \cup [0, 0.5[$

q_1 : $\{3.5\}$



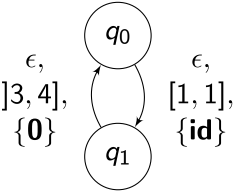


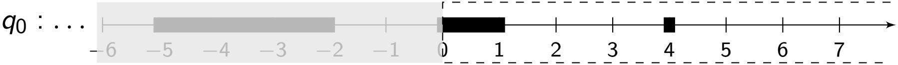
Regular Timed Interval

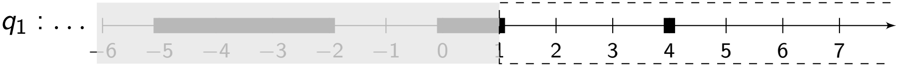
$\mathbf{U}_4$

$q_0 :\quad \{4\} \cup [0, 1[$

$q_1 :\quad \{4\}$





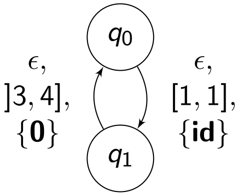


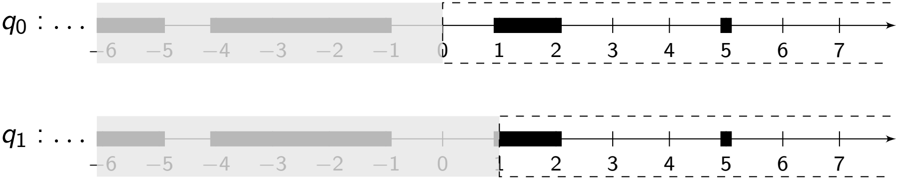
Regular Timed Interval

$\mathbf{U}_5$

$q_0 : \quad \{5\} \cup [1, 2[$

$q_1 : \quad \{5\} \cup [1, 2[$



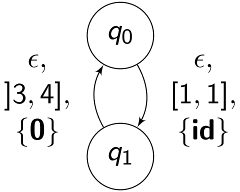


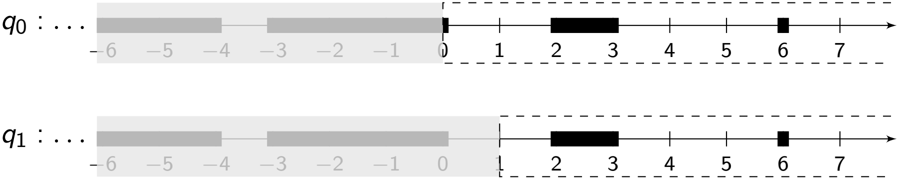
Regular Timed Interval

$\mathbf{U}_6$

q_0 : $\{6\} \cup [2,3[$

q_1 : $\{6\} \cup [2,3[$



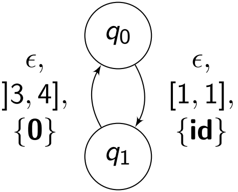


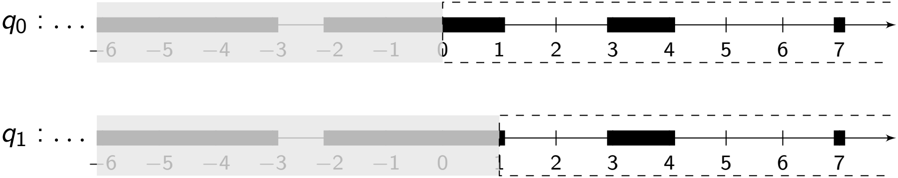
Regular Timed Interval

$\mathbf{U}_7$

$q_0 : \{7\} \cup [3,4[\cup [0,1[$

$q_1 : \{7\} \cup [3,4[$

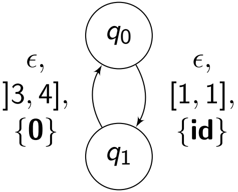




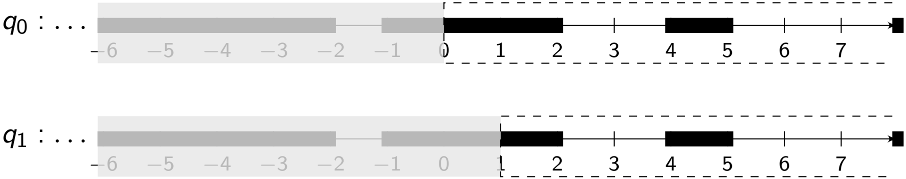
Regular Timed Interval

$\mathbf{U}_8$

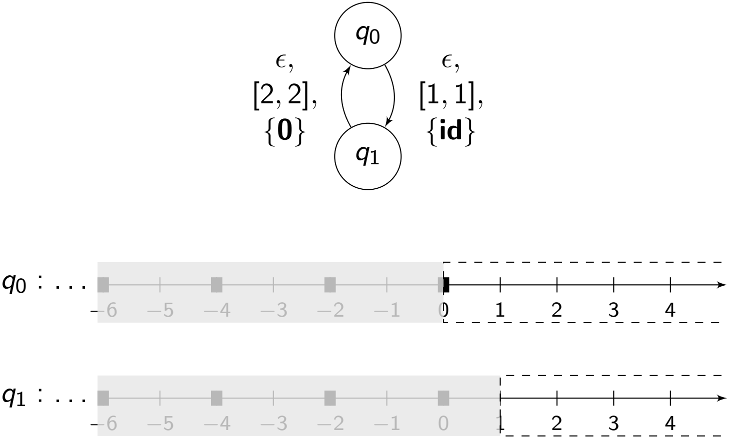
$q_0 : \{8\} \;\cup\; [4,5[\,\cup\, [0,2[$



$q_1 : \{8\} \;\cup\; [4,5[\,\cup [1,2[$



Regular Timed Interval



Regular Timed Interval

- A delay transition on the powerset automaton is equivalent to an addition on regular timed intervals once ϵ have been applied
- The operator ϵ is computable by sequential set theoretic operations
- Most of interval part of the computation can be precomputed
- Computing a delay can be done in constant time once the ϵ have been applied

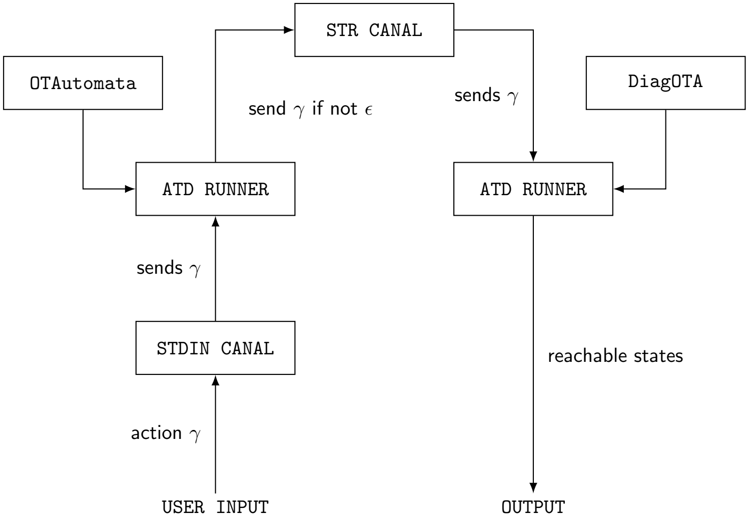
Step 3

- **Step 1 :** Determinization - General Powerset Construction
  - *Definition of an adapted model - Automata on Timed Structures*
  - *Definition of a general powerset construction for Automata on Timed Structures*
  - *Relations between the new powerset construction and previous determinization results*
- **Step 2 :** Diagnosis - A diagnoser for 1-clock timed automata
  - *Introduction of an adapted data structure - Regular Timed Interval*
  - *Powerset Construction + Regular Timed Interval = diagnoser*
- **Step 3 :** Implementation - The power of precomputation
  - *Implementation of the diagnoser of Tripakis.*
  - *Implementation of our diagnoser with some precomputation.*
  - *Comparison of the performance.*

DOTA

- $\approx$ 5500 lines of Python code
- Implementation of both methods
- Library for Regular Timed Intervals
- Several ways of using the diagnoser using channels

DOTA


STR CANAL
OTAutomata
DiagOTA
send γ if not ϵ
sends γ
ATD RUNNER
ATD RUNNER
sends γ
STDIN CANAL
reachable states
action γ
USER INPUT
OUTPUT

Benchmark

	Example2	Example3	Example4	Example6	Example8
#State/#Silent Trans	3/6	4/6	4/7	7/10	7/5
Precomputation Time	173.25s	0.38s	791.06s	11.01s	4.96s
Actions `DiagOTA`	0.014s	0.019s	0.029s	0.17s	0.15s
Actions `TripakisDOTA`	0.020s	0.078s	0.049s	0.26s	0.042
Ratio (actions)	0.73	0.25	0.59	0.64	3.71
Delays `DiagOTA`	0.000012s	0.0000011s	0.000011s	0.000011s	0.000012s
Delays `TripakisDOTA`	0.032s	0.057s	0.049s	0.30s	0.033s
Ratio (delays)	0.0004	0.0002	0.0002	0.00003	0.0004

Bench for 5 examples over 400 runs with 10 to 20 actions

Conclusion

- New quantitative model : Automata on Timed Structures
- General powerset construction
- Comparison with existing work
- Exploitation of the powerset construction in the context of diagnosis
- Comparison of our diagnoser with the diagnoser constructed by Tripakis

Conclusion

- New quantitative model : Automata on Timed Structures
- General powerset construction
- Comparison with existing work
- Exploitation of the powerset construction in the context of diagnosis
- Comparison of our diagnoser with the diagnoser constructed by Tripakis

Perspectives

- Improve DOTA and the benchmark
- Extend the diagnoser construction to n-clock timed automata
- Export classical theoretical result in the framework of automata on timed structures.

Thank You