The Cost of Punctuality

Patricia Bouyer, Nicolas Markey, Joël Ouaknine, James Worrell

LSV – CNRS & ENS de Cachan – France

Oxford University Computing Laboratory – UK

Motivation

Context: verification of timed systems towards linear-time timed temporal logics

Motivation

Context: verification of timed systems towards linear-time timed temporal logics

1. linear-time timed temporal logics: interesting for specifying properties of systems, but we cannot verify them! [AH93]

Motivation

Context: verification of timed systems towards linear-time timed temporal logics

1. linear-time timed temporal logics: interesting for specifying properties of systems, but we cannot verify them! [AH93]

2. MITL, a palliative to these negative results [AFH96]
   (MITL: disallows punctual constraints)

Motivation

Context: verification of timed systems towards linear-time timed temporal logics

1. linear-time timed temporal logics: interesting for specifying properties of systems, but we cannot verify them! [AH93]
2. MITL, a palliative to these negative results [AFH96]
   (MITL: disallows punctual constraints)

   $\rightarrow$ *punctuality is undecidable!*

Motivation

Context: verification of timed systems towards linear-time timed temporal logics

1. linear-time timed temporal logics: interesting for specifying properties of systems, but we cannot verify them! [AH93]
2. MITL, a palliative to these negative results [AFH96]
   (MITL: disallows punctual constraints)
   $\rightarrow$ *punctuality is undecidable!*
3. Safety-MTL: a decidable logic which partly allows punctuality [OW0{5,6}]

   However, it is non-primitive recursive!

Motivation

Context: verification of timed systems towards linear-time timed temporal logics

1. linear-time timed temporal logics: interesting for specifying properties of systems, but we cannot verify them! [AH93]
2. MITL, a palliative to these negative results [AFH96]
   (MITL: disallows punctual constraints)
   $\rightarrow$ *punctuality is undecidable!*
3. Safety-MTL: a decidable logic which partly allows punctuality [OW0{5,6}]

   However, it is non-primitive recursive!
4. we propose a tractable though powerful linear-time timed temporal logic which allows punctuality...

Metric Temporal Logic

MTL: Metric Temporal Logic [Koymans 1990]

$$\mathsf{MTL} \ni \varphi \ ::= \ a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \, \mathbf{U}_I \, \psi \mid \varphi \, \widetilde{\mathbf{U}}_I \, \varphi$$

where I is an interval with integral bounds

Metric Temporal Logic

MTL: Metric Temporal Logic [Koymans 1990]

$$\mathsf{MTL} \ni \varphi ::= a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi\, \mathbf{U}_I\, \psi \mid \varphi\, \widetilde{\mathbf{U}}_I\, \varphi$$

where I is an interval with integral bounds

We interpret MTL formulas over timed words (this is the so-called point-based semantics):

Metric Temporal Logic

MTL: Metric Temporal Logic [Koymans 1990]

$$\mathsf{MTL} \ni \varphi ::= a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \, \mathbf{U}_I \, \psi \mid \varphi \, \widetilde{\mathbf{U}}_I \, \varphi$$

where I is an interval with integral bounds

We interpret MTL formulas over timed words (this is the so-called point-based semantics):

We use classical shorthands, like $\mathbf{F}$, $\mathbf{G}$, $\mathbf{X}$, etc...

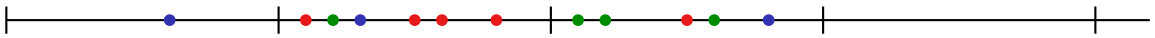
Metric Temporal Logic

MTL: Metric Temporal Logic [Koymans 1990]

$$\mathsf{MTL} \ni \varphi \ ::= \ a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \, \mathbf{U}_I \, \psi \mid \varphi \, \widetilde{\mathbf{U}}_I \, \varphi$$

where I is an interval with integral bounds

We interpret MTL formulas over timed words (this is the so-called point-based semantics):



We use classical shorthands, like $\mathbf{F}$, $\mathbf{G}$, $\mathbf{X}$, etc...

- $\mathbf{G}_{<2}\left(\bullet \rightarrow \mathbf{F}_{=1} \bullet\right)$

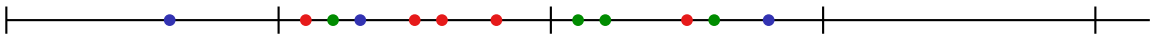
Metric Temporal Logic

MTL: Metric Temporal Logic [Koymans 1990]

$$\mathsf{MTL} \ni \varphi \ ::= \ a \ \mid \ \neg a \ \mid \ \varphi \vee \varphi \ \mid \ \varphi \wedge \varphi \ \mid \ \varphi \, \mathbf{U}_I \, \psi \ \mid \ \varphi \, \widetilde{\mathbf{U}}_I \, \varphi$$

where I is an interval with integral bounds

We interpret MTL formulas over timed words (this is the so-called point-based semantics):



We use classical shorthands, like $\mathbf{F}$, $\mathbf{G}$, $\mathbf{X}$, etc...

- $\mathbf{G}_{<2}\Big(\textcolor{red}{\bullet} \rightarrow \mathbf{F}_{=1}\,\textcolor{green}{\bullet}\Big)$
- $(\textcolor{blue}{\bullet}\,\mathbf{U}_{>3}\,\textcolor{red}{\bullet})\,\mathbf{U}_{[0,1]}\,(\mathbf{F}_{>1}\,\textcolor{green}{\bullet})$

Interesting Fragments of MTL

$$\mathsf{MTL} \ni \varphi \ ::= \ a \ \mid \ \neg a \ \mid \ \varphi \vee \varphi \ \mid \ \varphi \wedge \varphi \ \mid \ \varphi \, \mathbf{U}_I \, \varphi \ \mid \ \varphi \, \widetilde{\mathbf{U}}_I \, \varphi$$

MTL

Interesting Fragments of MTL

$$\text{LTL} \ni \varphi ::= a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \,\mathbf{U}\, \varphi \mid \varphi \,\widetilde{\mathbf{U}}\, \varphi$$

LTL $\hookrightarrow$ MTL

[Pnueli77]

Interesting Fragments of MTL

$$\mathsf{MITL} \ni \varphi \ ::= \ a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \, \mathbf{U}_{I} \, \varphi \mid \varphi \, \widetilde{\mathbf{U}}_{I} \, \varphi$$

with I non-singular, *i.e.*, with no "punctuality"

$$\mathsf{LTL} \hookrightarrow \mathsf{MITL} \hookrightarrow \mathsf{MTL}$$

[AFH96]

Interesting Fragments of MTL

$$\text{Bounded-MTL} \ni \varphi \;::=\; a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \,\mathbf{U}_I\, \varphi \mid \varphi \,\widetilde{\mathbf{U}}_I\, \varphi$$

with I bounded

$$\begin{tikzcd}
\text{Bounded-MTL} \arrow[rr, hook] & & \text{MTL} \\
\text{LTL} \arrow[r, hook] & \text{MITL} \arrow[ur, hook] &
\end{tikzcd}$$

Interesting Fragments of MTL

$$\text{Safety-MTL} \ni \varphi \ ::= \ a \ \mid \ \neg a \ \mid \ \varphi \vee \varphi \ \mid \ \varphi \wedge \varphi \ \mid \ \varphi \, \mathbf{U}_J \, \varphi \ \mid \ \varphi \, \widetilde{\mathbf{U}}_I \, \varphi$$

with J bounded

$$\begin{tikzcd}
 & \text{Safety-MTL} \arrow[dr, hook] & \\
\text{Bounded-MTL} \arrow[ur, hook] & & \text{MTL} \\
\text{LTL} \arrow[r, hook] & \text{MITL} \arrow[ur, hook] &
\end{tikzcd}$$

Bounded-MTL + Invariance $\subseteq$ Safety-MTL

[OW05]

Interesting Fragments of MTL

$$\text{Flat-MTL} \ni \varphi \ ::= \ a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \psi \, \mathbf{U}_I \, \varphi \mid \varphi \, \widetilde{\mathbf{U}}_I \, \psi$$

with I unbounded $\Rightarrow \psi \in$ LTL

$$\begin{tikzcd}
 & \text{Safety-MTL} \arrow[dr, hook] & \\
\text{Bounded-MTL} \arrow[ur, hook] & & \text{MTL} \\
\text{LTL} \arrow[r, hook] & \text{MITL} \arrow[ur, hook] &
\end{tikzcd}$$

Interesting Fragments of MTL

$$\text{coFlat-MTL} \ni \varphi \ ::= \ a \ \mid \ \neg a \ \mid \ \varphi \vee \varphi \ \mid \ \varphi \wedge \varphi \ \mid \ \varphi \, \mathbf{U}_I \, \psi \ \mid \ \psi \, \widetilde{\mathbf{U}}_I \, \varphi$$

with I unbounded $\Rightarrow \psi \in$ LTL

$$\begin{tikzcd}
 & \text{Safety-MTL} \arrow[dr, hook] & \\
\text{Bounded-MTL} \arrow[ur, hook] \arrow[r, hook] & \text{coFlat-MTL} \arrow[r, hook] & \text{MTL} \\
\text{LTL} \arrow[ur, hook] \arrow[r, hook] & \text{MITL} \arrow[ur, hook] &
\end{tikzcd}$$

Bounded-MTL + Invariance $\subseteq$ coFlat-MTL

Some Examples of Formulas

- $\mathbf{G}\left(\textit{request} \rightarrow \mathbf{F}_{[0,1]}\left(\textit{acquire} \wedge \mathbf{F}_{=1}\ \textit{release}\right)\right)$ is in coFlat-MTL, but neither in Bounded-MTL, nor in MITL.

Some Examples of Formulas

- $\mathbf{G}\left(\mathit{request} \rightarrow \mathbf{F}_{[0,1]}\left(\mathit{acquire} \wedge \mathbf{F}_{=1}\, \mathit{release}\right)\right)$ is in coFlat-MTL, but neither in Bounded-MTL, nor in MITL.

- $\varphi_n = \bullet \wedge \text{Double} \wedge \mathbf{G}_{[0,2^n)}\, \text{Double}$ where

$$\text{Double} = \left(\bullet \rightarrow \mathbf{F}_{=1}\left(\bullet \wedge \mathbf{X}_{<1}\, \bullet\right)\right) \wedge \left(\bullet \rightarrow \mathbf{F}_{=1}\left(\bullet \wedge \mathbf{X}_{<1}\, \bullet\right)\right)$$

  is in Bounded-MTL.

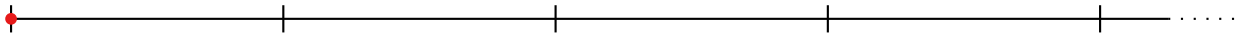
Some Examples of Formulas

- $\mathbf{G}\left(\textit{request} \rightarrow \mathbf{F}_{[0,1]}\left(\textit{acquire} \wedge \mathbf{F}_{=1}\ \textit{release}\right)\right)$ is in coFlat-MTL, but neither in Bounded-MTL, nor in MITL.

- $\varphi_n = \textcolor{red}{\bullet} \wedge \text{Double} \wedge \mathbf{G}_{[0,2^n)}\ \text{Double}$ where

$$\text{Double} = \left(\textcolor{red}{\bullet} \rightarrow \mathbf{F}_{=1}\left(\textcolor{red}{\bullet} \wedge \mathbf{X}_{<1}\ \textcolor{green}{\bullet}\right)\right) \wedge \left(\textcolor{green}{\bullet} \rightarrow \mathbf{F}_{=1}\left(\textcolor{red}{\bullet} \wedge \mathbf{X}_{<1}\ \textcolor{green}{\bullet}\right)\right)$$

  is in Bounded-MTL.



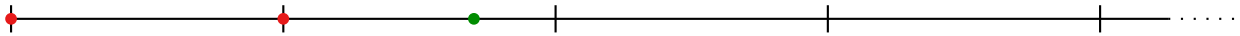
Some Examples of Formulas

- $\mathbf{G}\left(\mathit{request} \rightarrow \mathbf{F}_{[0,1]}\left(\mathit{acquire} \wedge \mathbf{F}_{=1}\,\mathit{release}\right)\right)$ is in coFlat-MTL, but neither in Bounded-MTL, nor in MITL.

- $\varphi_n = \textcolor{red}{\bullet} \wedge \textcolor{magenta}{\mathsf{Double}} \wedge \mathbf{G}_{[0,2^n)}\,\textcolor{magenta}{\mathsf{Double}}$ where

$$\textcolor{magenta}{\mathsf{Double}} = \left(\textcolor{red}{\bullet} \rightarrow \mathbf{F}_{=1}\left(\textcolor{red}{\bullet} \wedge \mathbf{X}_{<1}\,\textcolor{green}{\bullet}\right)\right) \wedge \left(\textcolor{green}{\bullet} \rightarrow \mathbf{F}_{=1}\left(\textcolor{red}{\bullet} \wedge \mathbf{X}_{<1}\,\textcolor{green}{\bullet}\right)\right)$$

is in Bounded-MTL.



Some Examples of Formulas

- $\mathbf{G}\left(\textit{request} \rightarrow \mathbf{F}_{[0,1]}\left(\textit{acquire} \wedge \mathbf{F}_{=1}\,\textit{release}\right)\right)$ is in coFlat-MTL, but neither in Bounded-MTL, nor in MITL.

- $\varphi_n = {\color{red}\bullet} \wedge \text{Double} \wedge \mathbf{G}_{[0,2^n)}\,\text{Double}$ where

$$\text{Double} = \left({\color{red}\bullet} \rightarrow \mathbf{F}_{=1}\left({\color{red}\bullet} \wedge \mathbf{X}_{<1}\,{\color{green}\bullet}\right)\right) \wedge \left({\color{green}\bullet} \rightarrow \mathbf{F}_{=1}\left({\color{red}\bullet} \wedge \mathbf{X}_{<1}\,{\color{green}\bullet}\right)\right)$$

  is in Bounded-MTL.

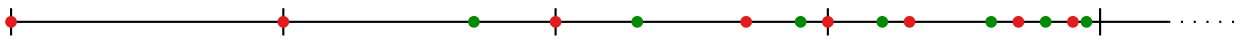
Some Examples of Formulas

- $\mathbf{G}\left(\textit{request} \rightarrow \mathbf{F}_{[0,1]}\left(\textit{acquire} \wedge \mathbf{F}_{=1}\, \textit{release}\right)\right)$ is in coFlat-MTL, but neither in Bounded-MTL, nor in MITL.

- $\varphi_n = {\color{red}\bullet} \wedge \text{Double} \wedge \mathbf{G}_{[0,2^n)}\, \text{Double}$ where

$$\text{Double} = \left({\color{red}\bullet} \rightarrow \mathbf{F}_{=1}\left({\color{red}\bullet} \wedge \mathbf{X}_{<1}\, {\color{green}\bullet}\right)\right) \wedge \left({\color{green}\bullet} \rightarrow \mathbf{F}_{=1}\left({\color{red}\bullet} \wedge \mathbf{X}_{<1}\, {\color{green}\bullet}\right)\right)$$

is in Bounded-MTL.



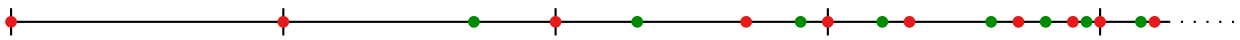
Some Examples of Formulas

- $\mathbf{G}\left(\textit{request} \rightarrow \mathbf{F}_{[0,1]}\left(\textit{acquire} \wedge \mathbf{F}_{=1}\,\textit{release}\right)\right)$ is in coFlat-MTL, but neither in Bounded-MTL, nor in MITL.

- $\varphi_n = \textcolor{red}{\bullet} \wedge \text{Double} \wedge \mathbf{G}_{[0,2^n)}\,\text{Double}$ where

$$\text{Double} = \left(\textcolor{red}{\bullet} \rightarrow \mathbf{F}_{=1}\left(\textcolor{red}{\bullet} \wedge \mathbf{X}_{<1}\,\textcolor{green}{\bullet}\right)\right) \wedge \left(\textcolor{green}{\bullet} \rightarrow \mathbf{F}_{=1}\left(\textcolor{red}{\bullet} \wedge \mathbf{X}_{<1}\,\textcolor{green}{\bullet}\right)\right)$$

is in Bounded-MTL.



Some Examples of Formulas

- $\mathbf{G}\left(\textit{request} \rightarrow \mathbf{F}_{[0,1]}\left(\textit{acquire} \wedge \mathbf{F}_{=1}\,\textit{release}\right)\right)$ is in coFlat-MTL, but neither in Bounded-MTL, nor in MITL.

- $\varphi_n = {\color{red}\bullet} \wedge \text{Double} \wedge \mathbf{G}_{[0,2^n)}\,\text{Double}$ where

$$\text{Double} = \left({\color{red}\bullet} \rightarrow \mathbf{F}_{=1}\left({\color{red}\bullet} \wedge \mathbf{X}_{<1}\,{\color{green}\bullet}\right)\right) \wedge \left({\color{green}\bullet} \rightarrow \mathbf{F}_{=1}\left({\color{red}\bullet} \wedge \mathbf{X}_{<1}\,{\color{green}\bullet}\right)\right)$$

  is in Bounded-MTL.

  $\rightarrow$ enforces in polynomial space a doubly exponential variability

Some Examples of Formulas (cont'd)

- Half $= \mathbf{F}_{=1}\, \mathtt{tt} \vee \mathbf{X}_{\leqslant 1}\, \mathbf{F}_{=1}\, \mathtt{tt}$

  $\longrightarrow$ may eliminate one over two actions

Some Examples of Formulas (cont'd)

- Half $= \mathbf{F}_{=1}\, \mathtt{tt} \vee \mathbf{X}_{\leqslant 1}\, \mathbf{F}_{=1}\, \mathtt{tt}$

  $\longrightarrow$ may eliminate one over two actions

- the formula

  $$\bullet \wedge \text{Double} \wedge \mathbf{G}_{[0,2^n)}\, \text{Double} \wedge \mathbf{G}_{[2^n,2^{n+1})}\, \text{Half} \wedge \mathbf{F}_{=2^{n+1}}\, (\bullet \wedge \mathbf{X}_{=1}\, \mathtt{tt})$$

  hence enforces exact doubling and halfing...

Complexity Results

Over infinite timed words:

	Model Checking	**Satisfiability**
LTL	PSPACE-C. [folklore]	PSPACE-C. [folklore]
MITL	EXPSPACE-C. [AFH96]	EXPSPACE-C. [AFH96]
Bounded-MTL		
Safety-MTL		Decidable [OW06]
coFlat-MTL		
MTL	Undec. [AH93,OW06]	Undec. [AH93,OW06]

Complexity Results

Over infinite timed words:

	Model Checking	**Satisfiability**
LTL	PSPACE-C. [folklore]	PSPACE-C. [folklore]
MITL	EXPSPACE-C. [AFH96]	EXPSPACE-C. [AFH96]
Bounded-MTL		
Safety-MTL	Non-Prim.-Rec. [forthc.]	Non-Elem. [forthc.]
coFlat-MTL		Undec. [OW06]
MTL	Undec. [AH93,OW06]	Undec. [AH93,OW06]

Complexity Results

Over infinite timed words:

	Model Checking	**Satisfiability**
LTL	PSPACE-C. [folklore]	PSPACE-C. [folklore]
MITL	EXPSPACE-C. [AFH96]	EXPSPACE-C. [AFH96]
Bounded-MTL	EXPSPACE-C.	EXPSPACE-C.
Safety-MTL	Non-Prim.-Rec. [forthc.]	Non-Elem. [forthc.]
coFlat-MTL	EXPSPACE-C.	Undec. [OW06]
MTL	Undec. [AH93,OW06]	Undec. [AH93,OW06]

An Example

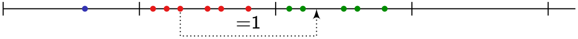
Assume one wants to verify formula

$$\mathbf{G}_{<2}\left(\textcolor{red}{\bullet} \to \mathbf{F}_{=1}\,\textcolor{green}{\bullet}\right)$$

An Example

Assume one wants to verify formula

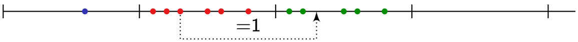
$$\mathbf{G}_{<2}\left(\textcolor{red}{\bullet} \rightarrow \mathbf{F}_{=1}\,\textcolor{green}{\bullet}\right)$$



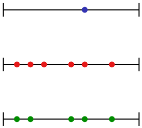
An Example

Assume one wants to verify formula

$$\mathbf{G}_{<2}\left(\bullet \rightarrow \mathbf{F}_{=1}\, \bullet\right)$$



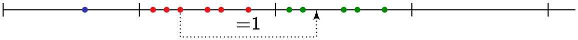
Offline, we stack all time units and use a sliding window:



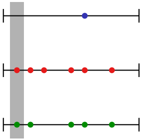
An Example

Assume one wants to verify formula

$$\mathbf{G}_{<2}\left(\textcolor{red}{\bullet} \rightarrow \mathbf{F}_{=1}\, \textcolor{green}{\bullet}\right)$$



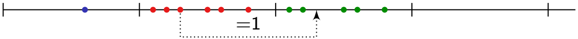
Offline, we stack all time units and use a sliding window:



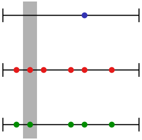
An Example

Assume one wants to verify formula

$$\mathbf{G}_{<2}\left({\color{red}\bullet} \rightarrow \mathbf{F}_{=1}\,{\color{green}\bullet}\right)$$



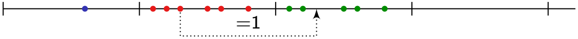
Offline, we stack all time units and use a sliding window:



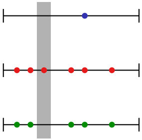
An Example

Assume one wants to verify formula

$$\mathbf{G}_{<2}\Big(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet\Big)$$



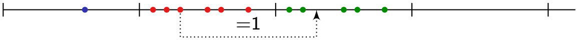
Offline, we stack all time units and use a sliding window:



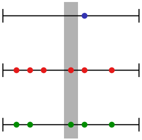
An Example

Assume one wants to verify formula

$$\mathbf{G}_{<2}\left(\textcolor{red}{\bullet} \rightarrow \mathbf{F}_{=1}\, \textcolor{green}{\bullet}\right)$$



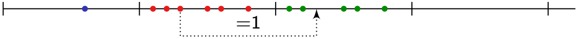
Offline, we stack all time units and use a sliding window:



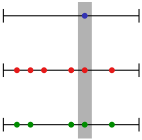
An Example

Assume one wants to verify formula

$$\mathbf{G}_{<2}\Big(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet\Big)$$



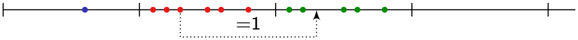
Offline, we stack all time units and use a sliding window:



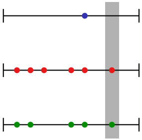
An Example

Assume one wants to verify formula

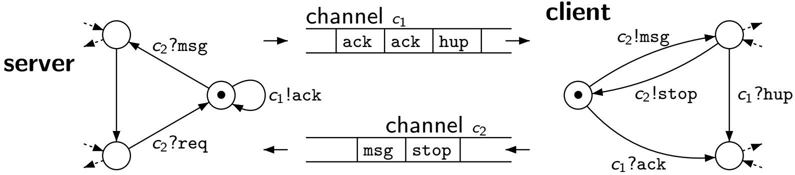
$$\mathbf{G}_{<2}\left(\textcolor{red}{\bullet} \rightarrow \mathbf{F}_{=1}\,\textcolor{green}{\bullet}\right)$$



Offline, we stack all time units and use a sliding window:



Channel Automata



NB: channels are FIFO...

Extended Channel Automata

We extend channel automata with:

- renaming (a letter can be replaced non-deterministically by another one);
- occurrence testing (check whether some letter appears on the channel).

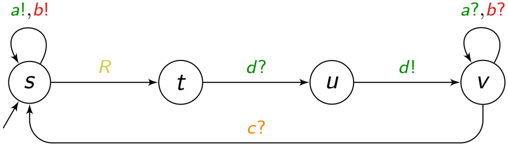
$\longrightarrow$ CAROT

Extended Channel Automata

We extend channel automata with:

- renaming (a letter can be replaced non-deterministically by another one);
- occurrence testing (check whether some letter appears on the channel).

$\rightarrow$ CAROT



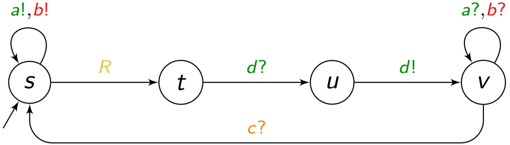
where R non-deterministically rename b to either b or c.

Extended Channel Automata

We extend channel automata with:

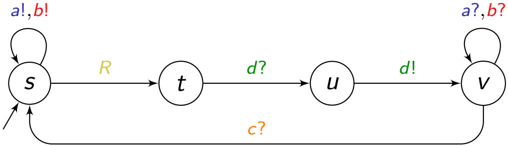
- renaming (a letter can be replaced non-deterministically by another one);
- occurrence testing (check whether some letter appears on the channel).

$\longrightarrow$ CAROT

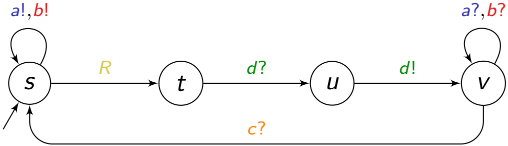


where R non-deterministically rename b to either b or c.

We will be interested in the reachability problem for CAROTs when we bound the number of cycles of the machine



where $R : b \mapsto b \vee c$

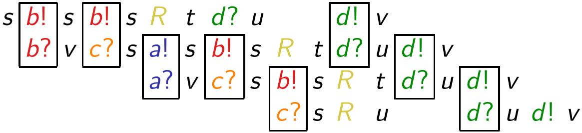


where $R : b \mapsto b \vee c$

Computation table, starting with d on the channel:

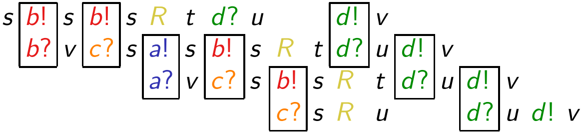
$$
\begin{array}{ccccccccccccccccccc}
s & b! & s & b! & s & R & t & d? & u & & & d! & v & & & & & & \\
 & b? & v & c? & s & a! & s & b! & s & R & t & d? & u & d! & v & & & & \\
 & & & & & a? & v & c? & s & b! & s & R & t & d? & u & d! & v & & \\
 & & & & & & & & & c? & s & R & u & & & d? & u & d! & v
\end{array}
$$

$$
\begin{array}{ccccccccccccccccccc}
s & b! & s & b! & s & R & t & d? & u & & & d! & v & & & & & & \\
 & b? & v & c? & s & a! & s & b! & s & R & t & d? & u & d! & v & & & & \\
 & & & & & a? & v & c? & s & b! & s & R & t & d? & u & d! & v & & \\
 & & & & & & & & & c? & s & R & u & & & d? & u & d! & v
\end{array}
$$



Computation table with sliding window:

s	$b!$	s	$b!$	s	R	t	$d?$	u	u	u	$d!$	v	v	v	v	v	v	v
v	$b?$	v	$c?$	s	$a!$	s	$b!$	s	R	t	$d?$	u	$d!$	v	v	v	v	v
v	v	v	v	v	$a?$	v	$c?$	s	$b!$	s	R	t	$d?$	u	$d!$	v	v	v
v	v	v	v	v	v	v	v	v	$c?$	s	R	u	u	u	$d?$	u	$d!$	v



Computation table with sliding window:

s	$b!$	s	$b!$	s	R	t	$d?$	u	u	u	$d!$	v	v	v	v	v	v	v
v	$b?$	v	$c?$	s	$a!$	s	$b!$	s	R	t	$d?$	u	$d!$	v	v	v	v	v
v	v	v	v	v	$a?$	v	$c?$	s	$b!$	s	R	t	$d?$	u	$d!$	v	v	v
v	v	v	v	v	v	v	v	v	$c?$	s	R	u	u	u	$d?$	u	$d!$	v

We need to store a window and some extra information for the renaming functions and the occurrence testing.

Theorem

The cycle-bounded reachability problem for CAROTs is solvable in polynomial space in the size of the channel automaton and polynomial space in the value of the cycle bound.

(Can guess and verify a computation table using polynomial space.)

Application to Timed Temporal Logics

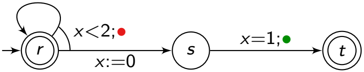
- Transform an MTL formula φ into an equivalent one-clock alternating timed automaton $\mathcal{A}_\varphi$ [OW05]

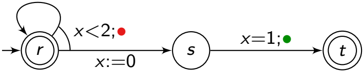
$$\mathbf{G}_{<2}\left(\textcolor{red}{\bullet} \rightarrow \mathbf{F}_{=1}\,\textcolor{green}{\bullet}\right)$$

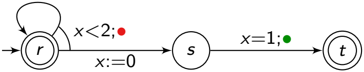
Application to Timed Temporal Logics

- Transform an MTL formula φ into an equivalent one-clock alternating timed automaton $\mathcal{A}_\varphi$ [OW05]

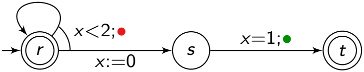
$$\mathbf{G}_{<2}\left(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet\right)$$



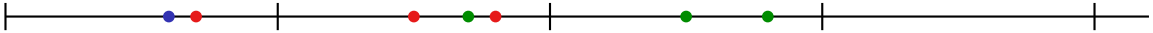

r
s
t
x<2;
x:=0
x=1;

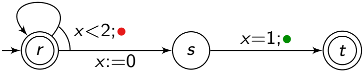


- See a behaviour of this automaton as the content of a FIFO channel

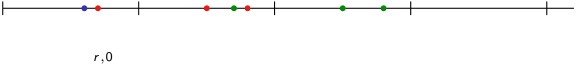


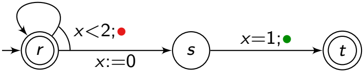
- See a behaviour of this automaton as the content of a FIFO channel



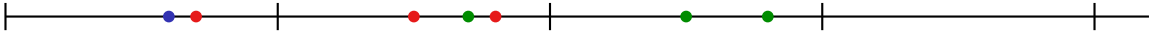


- See a behaviour of this automaton as the content of a FIFO channel

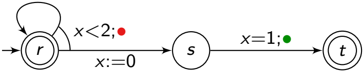




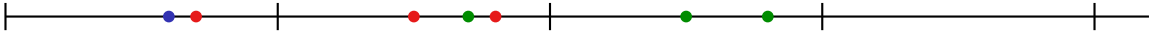
- See a behaviour of this automaton as the content of a FIFO channel

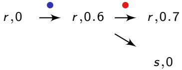


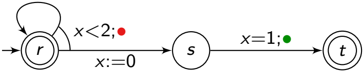
$r,0 \xrightarrow{\bullet} r,0.6$



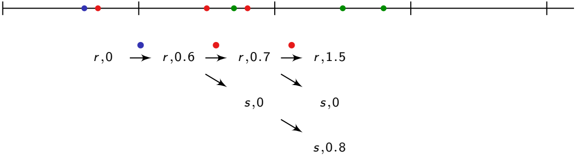
- See a behaviour of this automaton as the content of a FIFO channel

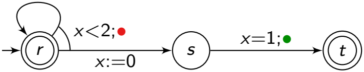




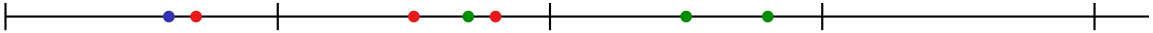


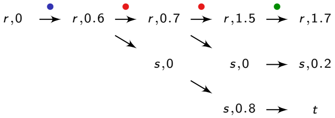
- See a behaviour of this automaton as the content of a FIFO channel

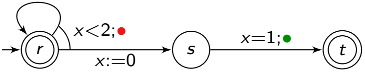




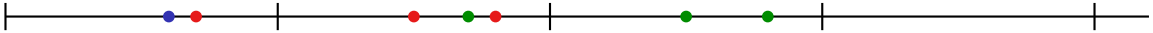
- See a behaviour of this automaton as the content of a FIFO channel

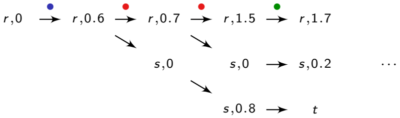


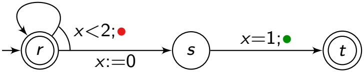




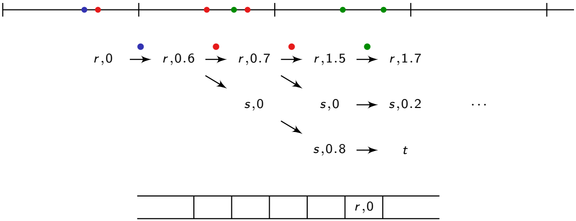
- See a behaviour of this automaton as the content of a FIFO channel

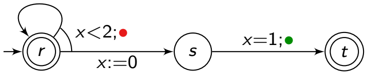




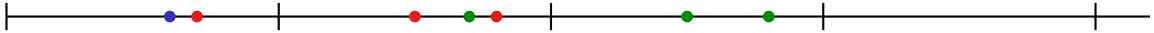


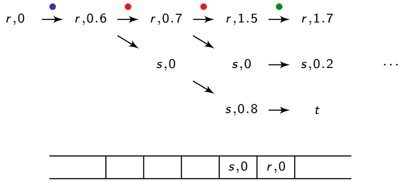
- See a behaviour of this automaton as the content of a FIFO channel

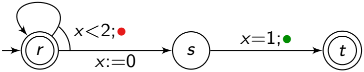




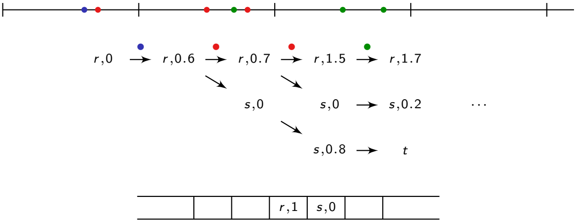
- See a behaviour of this automaton as the content of a FIFO channel

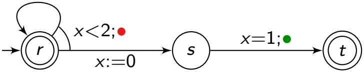




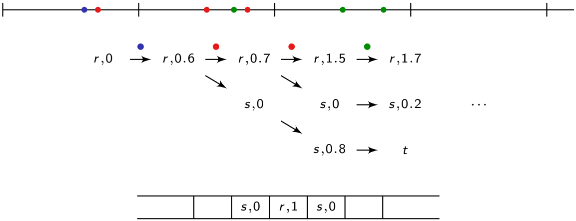


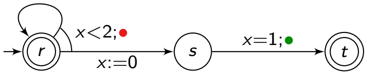
- See a behaviour of this automaton as the content of a FIFO channel



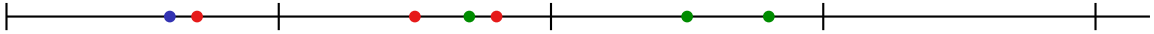


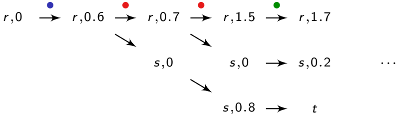
- See a behaviour of this automaton as the content of a FIFO channel

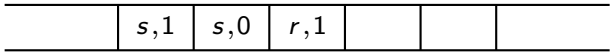


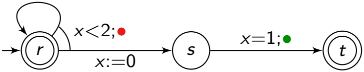


- See a behaviour of this automaton as the content of a FIFO channel

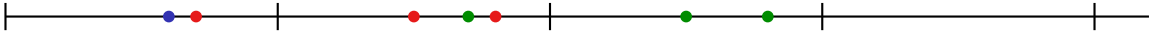


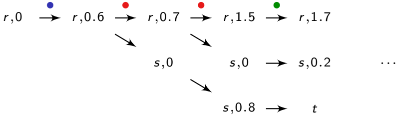




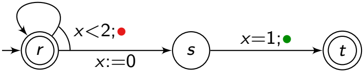


- See a behaviour of this automaton as the content of a FIFO channel

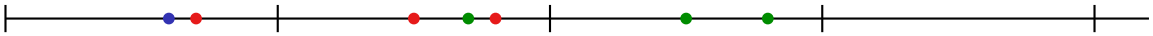


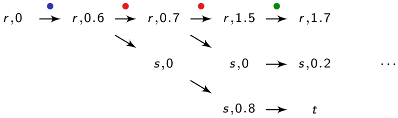


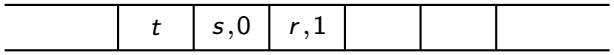
	t	$s,0$	$r,1$			



- See a behaviour of this automaton as the content of a FIFO channel



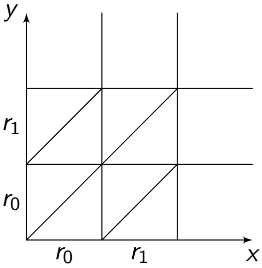




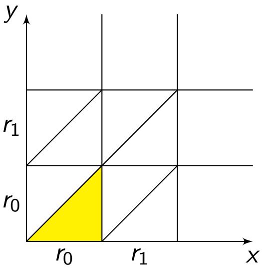
From MTL to CAROTs

Every formula φ can be transformed into a CAROT that “accepts” the models of φ.

A Digression on Timed Automata



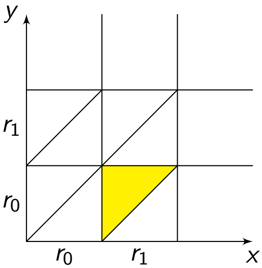
A Digression on Timed Automata



$x, y \in r_0$, $\{y\} < \{x\}$

	(y, r_0)	(x, r_0)	

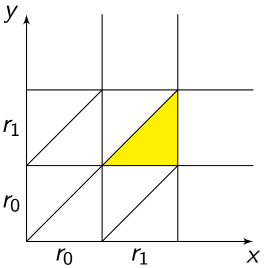
A Digression on Timed Automata



$x \in r_1$, $y \in r_0$, $\{x\} < \{y\}$

	(x, r_1)	(y, r_0)	

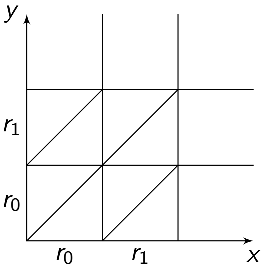
A Digression on Timed Automata



$x, y \in r_1$, $\{y\} < \{x\}$

	(y, r_1)	(x, r_1)	

A Digression on Timed Automata



The region graph can be simulated by a channel machine (with a single bounded channel).

Back to coFlat-MTL

We want to bound the number of cycles needed by the CAROT to achieve model-checking of coFlat-MTL, or more simply the satisfiability of Flat-MTL.

$$\text{Flat-MTL} \ni \varphi ::= a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \psi \, \mathbf{U}_I \, \varphi \mid \varphi \, \widetilde{\mathbf{U}}_I \, \psi$$

with I unbounded $\Rightarrow \psi \in$ LTL

Back to coFlat-MTL

We want to bound the number of cycles needed by the CAROT to achieve model-checking of coFlat-MTL, or more simply the satisfiability of Flat-MTL.

$$\text{Flat-MTL} \ni \varphi \ ::= \ a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \psi \, \mathbf{U}_I \, \varphi \mid \varphi \, \widetilde{\mathbf{U}}_I \, \psi$$

with I unbounded $\Rightarrow \psi \in$ LTL

A slice of the automaton:

$\{(a \, \mathbf{U} \, b, 1.6), ((p \, \mathbf{U} \, q) \, \mathbf{U} \, (\mathbf{F}_{=1} \, a), 2.5), (\mathbf{G}_{>3} \, (p \vee q), 4.1), ((\mathbf{F}_{<1} \, q) \, \mathbf{U}_{\leqslant 4} \, a, 3.9)\}$

Back to coFlat-MTL

We want to bound the number of cycles needed by the CAROT to achieve model-checking of coFlat-MTL, or more simply the satisfiability of Flat-MTL.

$$\text{Flat-MTL} \ni \varphi \ ::= \ a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \psi \, \mathbf{U}_I \, \varphi \mid \varphi \, \widetilde{\mathbf{U}}_I \, \psi$$

with I unbounded $\Rightarrow \psi \in$ LTL

A slice of the automaton:

$$\{(a \,\mathbf{U}\, b, 1.6), ((p \,\mathbf{U}\, q) \,\mathbf{U}\, (\mathbf{F}_{=1}\, a), 2.5), (\mathbf{G}_{>3}\, (p \vee q), 4.1), ((\mathbf{F}_{<1}\, q) \,\mathbf{U}_{\leqslant 4}\, a, 3.9)\}$$

Its encoding is:

$$\{((p \,\mathbf{U}\, q) \,\mathbf{U}\, (\mathbf{F}_{=1}\, a), \top), (\mathbf{G}_{>3}\, (p \vee q), \bot), ((\mathbf{F}_{<1}\, q) \,\mathbf{U}_{\leqslant 4}\, a, \top)\}$$

if we suppose the maximal constant is 4.

Back to coFlat-MTL

We want to bound the number of cycles needed by the CAROT to achieve model-checking of coFlat-MTL, or more simply the satisfiability of Flat-MTL.

$$\text{Flat-MTL} \ni \varphi ::= a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \psi \, \mathbf{U}_I \, \varphi \mid \varphi \, \widetilde{\mathbf{U}}_I \, \psi$$

with I unbounded $\Rightarrow \psi \in$ LTL

A slice of the automaton:

$$\{(a \, \mathbf{U} \, b, 1.6), ((p \, \mathbf{U} \, q) \, \mathbf{U} \, (\mathbf{F}_{=1} \, a), 2.5), (\mathbf{G}_{>3} \, (p \vee q), 4.1), ((\mathbf{F}_{<1} \, q) \, \mathbf{U}_{\leqslant 4} \, a, 3.9)\}$$

Its encoding is:

$$\{((p \, \mathbf{U} \, q) \, \mathbf{U} \, (\mathbf{F}_{=1} \, a), \top), (\mathbf{G}_{>3} \, (p \vee q), \bot), ((\mathbf{F}_{<1} \, q) \, \mathbf{U}_{\leqslant 4} \, a, \top)\}$$

if we suppose the maximal constant is 4.

$$\begin{cases} \top \rightsquigarrow \text{active} \\ \bot \rightsquigarrow \text{inactive} \end{cases}$$

A Ranking Function

We assume a linear order on pairs $(\psi, \mathbb{I})$ with ψ non-LTL modal subformula of φ, and $\mathbb{I} \in \{\top, \bot\}$ such that:

$$\begin{cases} (\psi, \bot) < (\psi, \top) \\ (\psi', \mathbb{I}') < (\psi, \mathbb{I}) \text{ if } \psi' \text{ subformula of } \psi \end{cases}$$

A Ranking Function

We assume a linear order on pairs $(\psi, \mathbb{I})$ with ψ non-LTL modal subformula of φ, and $\mathbb{I} \in \{\top, \bot\}$ such that:

$$\begin{cases} (\psi, \bot) < (\psi, \top) \\ (\psi', \mathbb{I}') < (\psi, \mathbb{I}) \text{ if } \psi' \text{ subformula of } \psi \end{cases}$$

$\mathsf{rank}(\gamma) = u.\alpha$ where α highest active subformula, and u all inactive subformulas (ordered with $<$) which are larger than α

A Ranking Function

We assume a linear order on pairs $(\psi, \mathbb{I})$ with ψ non-LTL modal subformula of φ, and $\mathbb{I} \in \{\top, \bot\}$ such that:

$$\begin{cases} (\psi, \bot) < (\psi, \top) \\ (\psi', \mathbb{I}') < (\psi, \mathbb{I}) \text{ if } \psi' \text{ subformula of } \psi \end{cases}$$

rank$(\gamma) = u.\alpha$ where α highest active subformula, and u all inactive subformulas (ordered with $<$) which are larger than α
$+$ we order the ranks with the lexicographic order

A Ranking Function

We assume a linear order on pairs $(\psi, \mathbb{I})$ with ψ non-LTL modal subformula of φ, and $\mathbb{I} \in \{\top, \bot\}$ such that:

$$\begin{cases} (\psi, \bot) < (\psi, \top) \\ (\psi', \mathbb{I}') < (\psi, \mathbb{I}) \text{ if } \psi' \text{ subformula of } \psi \end{cases}$$

$\mathsf{rank}(\gamma) = u.\alpha$ where α highest active subformula, and u all inactive subformulas (ordered with $<$) which are larger than α
$+$ we order the ranks with the lexicographic order

Properties

- if $\gamma \to \gamma'$, then $\mathsf{rank}(\gamma') \leqslant \mathsf{rank}(\gamma)$
- if γ is active (resp. inactive) and γ' is inactive (resp. active), and if $\gamma \to \gamma'$, then $\mathsf{rank}(\gamma') < \mathsf{rank}(\gamma)$
- if $\varrho : \gamma \to^* \gamma'$, and duration$(\varrho) > M$, then $\mathsf{rank}(\gamma') < \mathsf{rank}(\gamma)$

A Ranking Function

We assume a linear order on pairs $(\psi, \mathbb{I})$ with ψ non-LTL modal subformula of φ, and $\mathbb{I} \in \{\top, \bot\}$ such that:

$$\begin{cases} (\psi, \bot) < (\psi, \top) \\ (\psi', \mathbb{I}') < (\psi, \mathbb{I}) \text{ if } \psi' \text{ subformula of } \psi \end{cases}$$

$\mathsf{rank}(\gamma) = u.\alpha$ where α highest active subformula, and u all inactive subformulas (ordered with $<$) which are larger than α
$+$ we order the ranks with the lexicographic order

Properties

- if $\gamma \rightarrow \gamma'$, then $\mathsf{rank}(\gamma') \leqslant \mathsf{rank}(\gamma)$
- if γ is active (resp. inactive) and γ' is inactive (resp. active), and if $\gamma \rightarrow \gamma'$, then $\mathsf{rank}(\gamma') < \mathsf{rank}(\gamma)$
- if $\varrho : \gamma \rightarrow^* \gamma'$, and $\mathsf{duration}(\varrho) > M$, then $\mathsf{rank}(\gamma') < \mathsf{rank}(\gamma)$

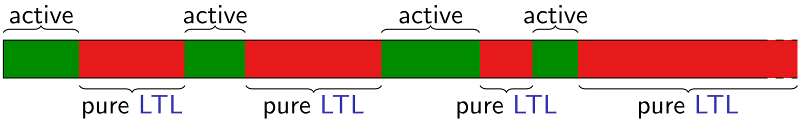
Hence, $\varrho = \varrho_0 \cdot \varrho_1 \cdot \ldots \varrho_{2n+1}$ with ϱ_{2i} (resp. ϱ_{2i+1}) active (resp. inactive) and $\sum_{i=0}^{n} \mathsf{duration}(\varrho_{2i}) \leqslant (M+1) \cdot |\varphi| \cdot 2^{|\varphi|}$

Model Checking coFlatMTL

Applying the previous decomposition of runs and the complexity of analyzing CAROTs, we get the following result:

Theorem

The model checking of coFlat-MTL is in EXPSPACE.



Hardness

Theorem

The satisfiability problem for Bounded-MTL is EXPSPACE-Hard.

Hardness

Theorem

The satisfiability problem for Bounded-MTL is EXPSPACE-Hard.

Encode the halting problem of an EXPSPACE Turing machine:

- generate a doubly exponential number of events in one time unit
- on the next time unit, non-deterministically guess a computation of the EXPSPACE Turing machine
- check it is correct (requires 2^n time units, one for each cell of the machine)
- half, and check that only one event remains

Conclusion

In this work, we have exhibited a subclass of MTL which:

- contains punctual constraints,
- contains invariance,
- is tractable in theory.

Conclusion

In this work, we have exhibited a subclass of MTL which:

- contains punctual constraints,
- contains invariance,
- is tractable in theory.

What needs to be done:

- check tractability in practice,
- extend to continuous semantics.