Model-Checking Timed Temporal Logics

Patricia Bouyer

LSV – CNRS & ENS de Cachan – France

Based on joint works with Fabrice Chevalier, Nicolas Markey,
Joël Ouaknine and James Worrell

Outline

Model-checking

system:

property:

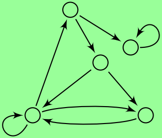
$\Rightarrow$

Model-checking

system:

property:

$\Rightarrow$

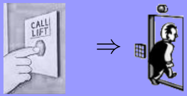


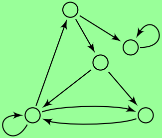
$\mathbf{G}\,(\texttt{request} \rightarrow \mathbf{F}\,\texttt{grant})$

Model-checking

system:

property:

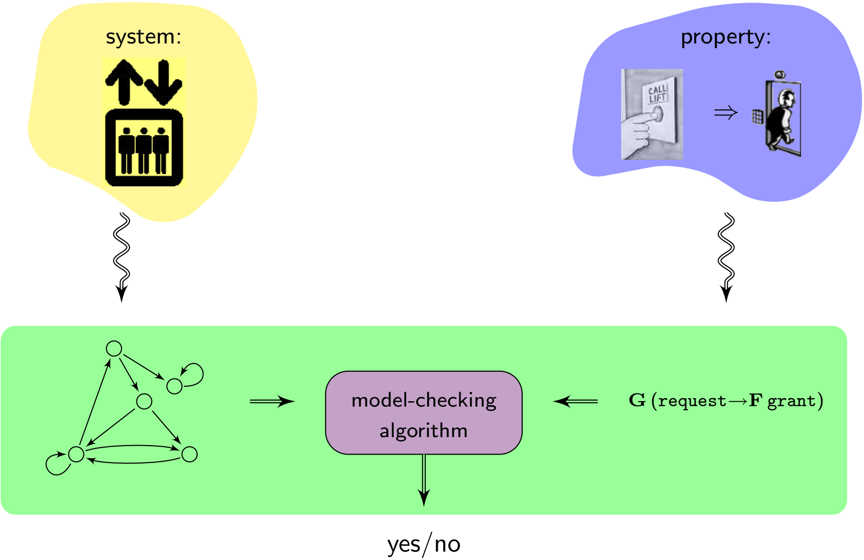




model-checking algorithm

$\mathbf{G}\,(\mathtt{request} \rightarrow \mathbf{F}\,\mathtt{grant})$

Model-checking



The untimed (linear-time) framework

Linear-time temporal logic [Pnu77]

$$\mathsf{LTL} \ni \varphi ::= p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

[Pnu77] Pnueli. The temporal logic of programs (FOCS'77).

The untimed (linear-time) framework

Linear-time temporal logic [Pnu77]

$$\mathsf{LTL} \ni \varphi ::= p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

$\models \quad \mathbf{X}$

[Pnu77] Pnueli. The temporal logic of programs (FOCS'77).

The untimed (linear-time) framework

Linear-time temporal logic [Pnu77]

$$\mathsf{LTL} \ni \varphi ::= p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

 $\models \quad \mathbf{X}\,\bullet$

[Pnu77] Pnueli. The temporal logic of programs (FOCS'77).

The untimed (linear-time) framework

Linear-time temporal logic [Pnu77]

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

$\models \quad \mathbf{X}\,\bullet$

$\models \quad \bullet\,\mathbf{U}\,\bullet$

[Pnu77] Pnueli. The temporal logic of programs (FOCS'77).

The untimed (linear-time) framework

Linear-time temporal logic [Pnu77]

$$\mathsf{LTL} \ni \varphi ::= p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

$\models \quad \mathbf{X}\,\bullet$

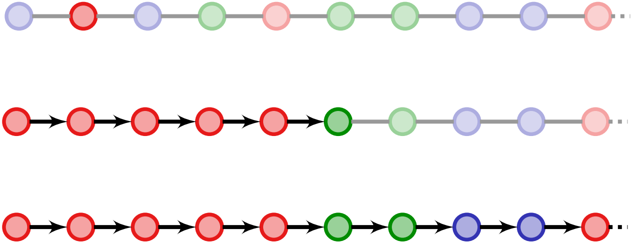
$\models \quad \bullet\,\mathbf{U}\,\bullet$

[Pnu77] Pnueli. The temporal logic of programs (FOCS'77).

The untimed (linear-time) framework

Linear-time temporal logic [Pnu77]

$$\mathsf{LTL} \ni \varphi ::= p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$



$\models \quad \mathbf{X}\,\textcolor{red}{\bullet}$

$\models \quad \textcolor{red}{\bullet}\,\mathbf{U}\,\textcolor{green}{\bullet}$

$\models \quad \mathbf{F}\,\textcolor{green}{\bullet} \;\equiv\; \mathtt{tt}\,\mathbf{U}\,\textcolor{green}{\bullet}$

[Pnu77] Pnueli. The temporal logic of programs (FOCS'77).

The untimed (linear-time) framework

Linear-time temporal logic [Pnu77]

$$\mathsf{LTL} \ni \varphi \ ::= \ p \ | \ \varphi \wedge \varphi \ | \ \varphi \vee \varphi \ | \ \neg\varphi \ | \ \mathbf{X}\,\varphi \ | \ \varphi\,\mathbf{U}\,\varphi$$

 $\models \quad \mathbf{X}\,\bullet$

 $\models \quad \bullet\,\mathbf{U}\,\bullet$

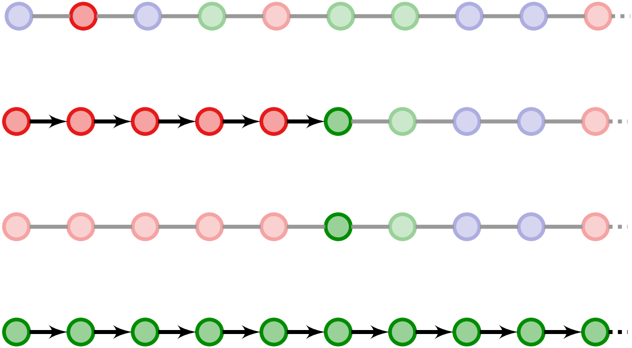
 $\models \quad \mathbf{F}\,\bullet \ \equiv \ \mathtt{tt}\,\mathbf{U}\,\bullet$

[Pnu77] Pnueli. The temporal logic of programs (FOCS'77).

The untimed (linear-time) framework

Linear-time temporal logic [Pnu77]

$$\mathsf{LTL} \ni \varphi ::= p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$



$$\models \quad \mathbf{X}\,\bullet$$

$$\models \quad \bullet\,\mathbf{U}\,\bullet$$

$$\models \quad \mathbf{F}\,\bullet \;\equiv\; \mathtt{tt}\,\mathbf{U}\,\bullet$$

$$\models \quad \mathbf{G}\,\bullet \;\equiv\; \neg\mathbf{F}\,\neg\bullet$$

[Pnu77] Pnueli. The temporal logic of programs (FOCS'77).

The untimed (linear-time) framework

Linear-time temporal logic [Pnu77]

$$\mathsf{LTL} \ni \varphi ::= p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

- response property:

$$\mathbf{G}\,(\bullet \rightarrow \mathbf{F}\,\bullet)$$

[Pnu77] Pnueli. The temporal logic of programs (FOCS'77).

The untimed (linear-time) framework

Linear-time temporal logic [Pnu77]

$$\mathsf{LTL} \ni \varphi ::= p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

- response property:

$$\mathbf{G}\,(\bullet \rightarrow \mathbf{F}\,\bullet)$$

- liveness property:

$$\mathbf{G}\,\mathbf{F}\,\bullet$$

[Pnu77] Pnueli. The temporal logic of programs (FOCS'77).

The untimed (linear-time) framework

Linear-time temporal logic [Pnu77]

$$\mathsf{LTL} \ni \varphi ::= p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

- response property:

$$\mathbf{G}\,({\color{red}\bullet} \to \mathbf{F}\,{\color{green}\bullet})$$

- liveness property:

$$\mathbf{G}\,\mathbf{F}\,{\color{green}\bullet}$$

- safety property:

$$\mathbf{G}\,\neg{\color{red}\bullet}$$

[Pnu77] Pnueli. The temporal logic of programs (FOCS'77).

The untimed (linear-time) framework

Linear-time temporal logic [Pnu77]

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

- response property:

$$\mathbf{G}\,(\bullet \rightarrow \mathbf{F}\,\bullet)$$

- liveness property:

$$\mathbf{G}\,\mathbf{F}\,\bullet$$

- safety property:

$$\mathbf{G}\,\neg\bullet$$

- a more complex property:

$$(\bullet \wedge (\mathbf{F}\,\bullet \vee \mathbf{G}\,\bullet))\,\mathbf{U}\,\bullet$$

[Pnu77] Pnueli. The temporal logic of programs (FOCS'77).

Adding timing requirements

- Need for timed models
  - the behaviour of most systems depends on time;
  - faithful modelling has to take time into account.

☞ timed automata, time(d) Petri nets, timed process algebras...

Adding timing requirements

- Need for timed models
  - the behaviour of most systems depends on time;
  - faithful modelling has to take time into account.

☞ timed automata, time(d) Petri nets, timed process algebras...

- Need for timed specification languages
  - the behaviour of most systems depends on time;
  - untimed specifications are not sufficient
    (for instance, bounded response timed, etc...)

☞ TCTL, MTL, TPTL, timed μ-calculus...

Outline

Metric Temporal Logic (MTL)

[Koy90]

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

where I is an interval with integral bounds.

[Koy90] Koymans. Specifying real-time properties with metric temporal logic (Real-time systems, 1990).

Metric Temporal Logic (MTL)

[Koy90]

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

where I is an interval with integral bounds.

- This is a timed extension of LTL

[Koy90] Koymans. Specifying real-time properties with metric temporal logic (Real-time systems, 1990).

Metric Temporal Logic (MTL)

[Koy90]

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

where I is an interval with integral bounds.

- This is a timed extension of LTL
- Can be interpreted over timed words, or over signals
  - this distinction is fundamental

[Koy90] Koymans. Specifying real-time properties with metric temporal logic (Real-time systems, 1990).

Metric Temporal Logic (MTL)

[Koy90]

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

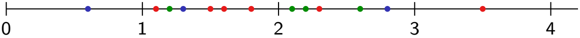
where I is an interval with integral bounds.

- This is a timed extension of LTL
- Can be interpreted over timed words, or over signals
  - this distinction is fundamental
- Can be interpreted over finite or infinite behaviours
  - this distinction is fundamental

[Koy90] Koymans. Specifying real-time properties with metric temporal logic (Real-time systems, 1990).

The pointwise semantics

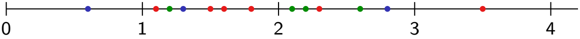
MTL formulas are interpreted over timed words:



$$(\bullet, .6)(\bullet, 1.1)(\bullet, 1.2)(\bullet, 1.3)\ldots$$

The pointwise semantics

MTL formulas are interpreted over timed words:

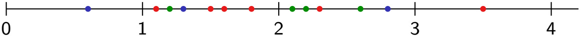


$$(\bullet, .6)(\bullet, 1.1)(\bullet, 1.2)(\bullet, 1.3)\dots$$

☞ the system is observed only when actions happen

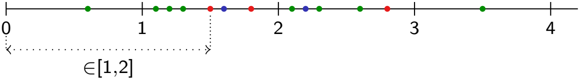
The pointwise semantics

MTL formulas are interpreted over timed words:



$$(\bullet, .6)(\bullet, 1.1)(\bullet, 1.2)(\bullet, 1.3)\ldots$$

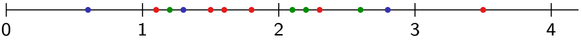
☞ the system is observed only when actions happen



$\models \bullet \ \mathbf{U}_{[1,2]} \bullet$

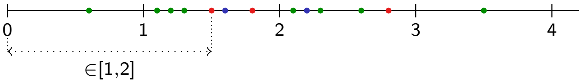
The pointwise semantics

MTL formulas are interpreted over timed words:

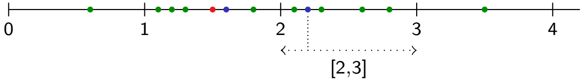


$$(\bullet, .6)(\bullet, 1.1)(\bullet, 1.2)(\bullet, 1.3)\ldots$$

☞ the system is observed only when actions happen



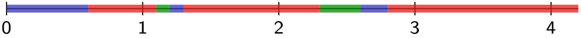
$\models \bullet\ \mathbf{U}_{[1,2]} \bullet$



$\not\models \mathbf{G}_{[2,3]} \bullet$

The continuous semantics

MTL formulas are interpreted over (finitely variable) signals:

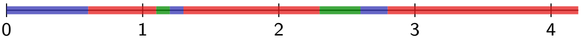


$t \in [0,.6] \mapsto \blacksquare$ $\qquad$ $t \in (.6,1.1) \mapsto \blacksquare$ $\qquad$ $t \in [1.1,1.2) \mapsto \blacksquare$

$\ldots$

The continuous semantics

MTL formulas are interpreted over (finitely variable) signals:



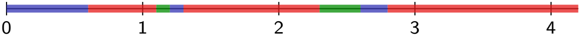
$t \in [0,.6] \mapsto$ ■ $\qquad t \in (.6,1.1) \mapsto$ ■ $\qquad t \in [1.1,1.2) \mapsto$ ■

...

☞ the system is observed continuously

The continuous semantics

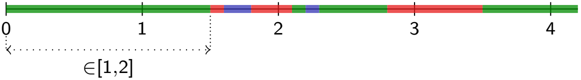
MTL formulas are interpreted over (finitely variable) signals:



$t \in [0,.6] \mapsto$ ■ $t \in (.6,1.1) \mapsto$ ■ $t \in [1.1,1.2) \mapsto$ ■

...

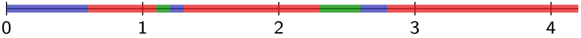
☞ the system is observed continuously



$\models \bullet\, \mathbf{U}_{[1,2]} \bullet$

The continuous semantics

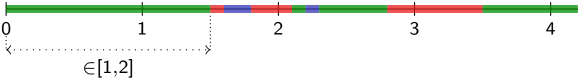
MTL formulas are interpreted over (finitely variable) signals:



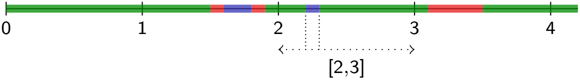
$t \in [0,.6] \mapsto$ ■ $\quad t \in (.6,1.1) \mapsto$ ■ $\quad t \in [1.1,1.2) \mapsto$ ■

…

☞ the system is observed continuously



$\models \bullet\ \mathbf{U}_{[1,2]} \bullet$



$\not\models \mathbf{G}_{[2,3]} \bullet$

Some examples

- "Every problem is followed within 56 time units by an alarm"

$$\mathbf{G}\,(\texttt{problem} \rightarrow \mathbf{F}_{\leqslant 56}\,\texttt{alarm})$$

Some examples

- "Every problem is followed within 56 time units by an alarm"

$$\mathbf{G}\,(\texttt{problem} \rightarrow \mathbf{F}_{\leqslant 56}\,\texttt{alarm})$$

- "Each time there is a problem, it is either repaired within the next 15 time units, or an alarm rings during 3 time units 12 time units later"

$$\mathbf{G}\,(\texttt{problem} \rightarrow (\mathbf{F}_{\leqslant 15}\,\texttt{repair} \vee \mathbf{G}_{[12,15)}\,\texttt{alarm}))$$

Some further extensions

- Timed Propositional Temporal Logic (TPTL) [AH89]

TPTL = LTL + clock variables + clock constraints

[AH89] Alur, Henzinger. A really temporal logic (FOCS'89).

Some further extensions

- Timed Propositional Temporal Logic (TPTL) [AH89]

TPTL = LTL + clock variables + clock constraints

$$\mathbf{G}\,(\texttt{problem} \rightarrow \mathbf{F}_{\leqslant 56}\,\texttt{alarm}) \quad \equiv \quad \mathbf{G}\,(\texttt{problem} \rightarrow x.\mathbf{F}\,(\texttt{alarm} \wedge x \leqslant 56))$$

[AH89] Alur, Henzinger. A really temporal logic (FOCS'89).

Some further extensions

- Timed Propositional Temporal Logic (TPTL) [AH89]

TPTL = LTL + clock variables + clock constraints

$$\mathbf{G}\,(\mathtt{problem} \rightarrow \mathbf{F}_{\leqslant 56}\,\mathtt{alarm}) \quad \equiv \quad \mathbf{G}\,(\mathtt{problem} \rightarrow x.\mathbf{F}\,(\mathtt{alarm} \wedge x \leqslant 56))$$

$$\mathbf{G}\,(\mathtt{problem} \rightarrow x.\mathbf{F}\,(\mathtt{alarm} \wedge \mathbf{F}\,(\mathtt{failsafe} \wedge x \leqslant 56)))$$

[AH89] Alur, Henzinger. A really temporal logic (FOCS'89).

Some further extensions

- Timed Propositional Temporal Logic (TPTL) [AH89]

TPTL = LTL + clock variables + clock constraints

$$\mathbf{G}\,(\texttt{problem} \rightarrow \mathbf{F}_{\leqslant 56}\,\texttt{alarm}) \quad \equiv \quad \mathbf{G}\,(\texttt{problem} \rightarrow x.\mathbf{F}\,(\texttt{alarm} \wedge x \leqslant 56))$$

$$\mathbf{G}\,(\texttt{problem} \rightarrow x.\mathbf{F}\,(\texttt{alarm} \wedge \mathbf{F}\,(\texttt{failsafe} \wedge x \leqslant 56)))$$

- MTL+Past: add past-time modalities [AH92]

[AH89] Alur, Henzinger. A really temporal logic (FOCS'89).
[AH92] Alur, Henzinger. Back to the future: towards a theory of timed regular languages (FOCS'92).

Some further extensions

- Timed Propositional Temporal Logic (TPTL) [AH89]

TPTL = LTL + clock variables + clock constraints

$$\mathbf{G}\,(\texttt{problem} \rightarrow \mathbf{F}_{\leqslant 56}\,\texttt{alarm}) \quad \equiv \quad \mathbf{G}\,(\texttt{problem} \rightarrow x.\mathbf{F}\,(\texttt{alarm} \wedge x \leqslant 56))$$

$$\mathbf{G}\,(\texttt{problem} \rightarrow x.\mathbf{F}\,(\texttt{alarm} \wedge \mathbf{F}\,(\texttt{failsafe} \wedge x \leqslant 56)))$$

- MTL+Past: add past-time modalities [AH92]

$$\mathbf{G}\,(\texttt{alarm} \rightarrow \mathbf{F}^{-1}_{\leqslant 56}\,\texttt{problem})$$

[AH89] Alur, Henzinger. A really temporal logic (FOCS'89).
[AH92] Alur, Henzinger. Back to the future: towards a theory of timed regular languages (FOCS'92).

A note on the expressiveness

Theorem

LTL+Past is as expressive as LTL [Kam68,GPSS80].

[Kam68] Kamp. Tense logic and the theory of linear order (PhD Thesis UCLA 1968).
[GPSS80] Gabbay, Pnueli, Shelah, Stavi. On the temporal analysis of fairness (POPL'80).

A note on the expressiveness

Theorem

LTL+Past is as expressive as LTL [Kam68,GPSS80].

Theorem

MTL is strictly less expressive than MTL+Past and TPTL [BCM05].

[Kam68] Kamp. Tense logic and the theory of linear order (PhD Thesis UCLA 1968).
[GPSS80] Gabbay, Pnueli, Shelah, Stavi. On the temporal analysis of fairness (POPL'80).
[BCM05] Bouyer, Chevalier, Markey. On the expressiveness of MTL and TPTL (FSTTCS'05).

A note on the expressiveness

Theorem

LTL+Past is as expressive as LTL [Kam68,GPSS80].

Theorem

MTL is strictly less expressive than MTL+Past and TPTL [BCM05].

Conjecture in 1990: the TPTL formula

$$\mathbf{G}\,(\bullet \rightarrow x.\mathbf{F}\,(\bullet \wedge \mathbf{F}\,(\bullet \wedge x \leqslant 2)))$$

cannot be expressed in MTL.

[Kam68] Kamp. Tense logic and the theory of linear order (PhD Thesis UCLA 1968).
[GPSS80] Gabbay, Pnueli, Shelah, Stavi. On the temporal analysis of fairness (POPL'80).
[BCM05] Bouyer, Chevalier, Markey. On the expressiveness of MTL and TPTL (FSTTCS'05).

A note on the expressiveness

Theorem

LTL+Past is as expressive as LTL [Kam68,GPSS80].

Theorem

MTL is strictly less expressive than MTL+Past and TPTL [BCM05].

Conjecture in 1990: the TPTL formula

$$\mathbf{G}\,(\bullet \rightarrow x.\mathbf{F}\,(\bullet \wedge \mathbf{F}\,(\bullet \wedge x \leqslant 2)))$$

cannot be expressed in MTL.

- This is true in the pointwise semantics.

[Kam68] Kamp. Tense logic and the theory of linear order (PhD Thesis UCLA 1968).
[GPSS80] Gabbay, Pnueli, Shelah, Stavi. On the temporal analysis of fairness (POPL'80).
[BCM05] Bouyer, Chevalier, Markey. On the expressiveness of MTL and TPTL (FSTTCS'05).

A note on the expressiveness

Theorem

LTL+Past is as expressive as LTL [Kam68,GPSS80].

Theorem

MTL is strictly less expressive than MTL+Past and TPTL [BCM05].

Conjecture in 1990: the TPTL formula

$$\mathbf{G}\,(\bullet \rightarrow x.\mathbf{F}\,(\bullet \wedge \mathbf{F}\,(\bullet \wedge x \leqslant 2)))$$

cannot be expressed in MTL.

- This is true in the pointwise semantics.
- This is wrong in the continuous semantics!

[Kam68] Kamp. Tense logic and the theory of linear order (PhD Thesis UCLA 1968).
[GPSS80] Gabbay, Pnueli, Shelah, Stavi. On the temporal analysis of fairness (POPL'80).
[BCM05] Bouyer, Chevalier, Markey. On the expressiveness of MTL and TPTL (FSTTCS'05).

The TPTL formula

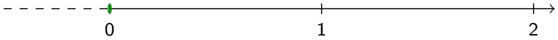
$$\mathbf{G}\left(\bullet \rightarrow x.\mathbf{F}\left(\bullet \wedge \mathbf{F}\left(\bullet \wedge x \leqslant 2\right)\right)\right)$$

can be expressed in MTL in the continuous semantics

The TPTL formula

$$\mathbf{G}\left(\bullet \rightarrow x.\mathbf{F}\left(\bullet \wedge \mathbf{F}\left(\bullet \wedge x \leqslant 2\right)\right)\right)$$

can be expressed in MTL in the continuous semantics

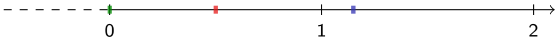


$$\mathbf{G}\ \bullet \rightarrow \left\{ \right.$$

The TPTL formula

$$\mathbf{G}\left(\bullet \rightarrow x.\mathbf{F}\left(\bullet \wedge \mathbf{F}\left(\bullet \wedge x \leqslant 2\right)\right)\right)$$

can be expressed in MTL in the continuous semantics

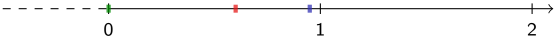


$$\mathbf{G}\ \bullet\ \rightarrow \left\{ \begin{array}{l} \mathbf{F}_{\leqslant 1}\ \bullet\ \wedge\ \mathbf{F}_{[1,2]}\ \bullet \\ \\ \\ \end{array} \right.$$

The TPTL formula

$$\mathbf{G}\left({\color{green}\bullet} \rightarrow x.\mathbf{F}\left({\color{red}\bullet} \wedge \mathbf{F}\left({\color{blue}\bullet} \wedge x \leqslant 2\right)\right)\right)$$

can be expressed in MTL in the continuous semantics

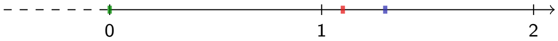


$$\mathbf{G}\ {\color{green}\bullet} \ \rightarrow \ \left\{ \begin{array}{l} \quad \mathbf{F}_{\leqslant 1}\ {\color{red}\bullet} \ \wedge \ \mathbf{F}_{[1,2]}\ {\color{blue}\bullet} \\ \vee \\ \quad \mathbf{F}_{\leqslant 1}\left({\color{red}\bullet} \ \wedge \ \mathbf{F}_{\leqslant 1}\,{\color{blue}\bullet}\right) \\ \\ \end{array} \right.$$

The TPTL formula

$$\mathbf{G}\,(\textcolor{green}{\bullet} \rightarrow x.\mathbf{F}\,(\textcolor{red}{\bullet} \wedge \mathbf{F}\,(\textcolor{blue}{\bullet} \wedge x \leqslant 2)))$$

can be expressed in MTL in the continuous semantics

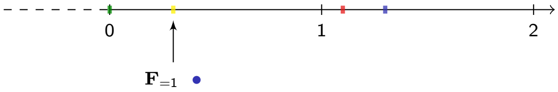


$$\mathbf{G}\ \textcolor{green}{\bullet}\ \rightarrow \left\{ \begin{array}{ll} & \mathbf{F}_{\leqslant 1}\ \textcolor{red}{\bullet}\ \wedge\ \mathbf{F}_{[1,2]}\ \textcolor{blue}{\bullet} \\ \vee & \\ & \mathbf{F}_{\leqslant 1}\,(\ \textcolor{red}{\bullet}\ \wedge\ \mathbf{F}_{\leqslant 1}\,\textcolor{blue}{\bullet}\ ) \\ \vee & \\ & \end{array} \right.$$

The TPTL formula

$$\mathbf{G}\,(\bullet \rightarrow x.\mathbf{F}\,(\bullet \wedge \mathbf{F}\,(\bullet \wedge x \leqslant 2)))$$

can be expressed in MTL in the continuous semantics

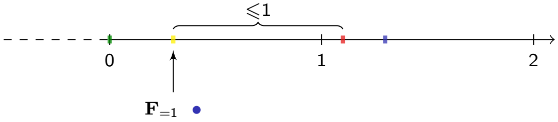


$$\mathbf{G}\ \bullet\ \rightarrow \begin{cases} \quad \mathbf{F}_{\leqslant 1}\ \bullet\ \wedge\ \mathbf{F}_{[1,2]}\ \bullet \\ \vee \\ \quad \mathbf{F}_{\leqslant 1}\,(\ \bullet\ \wedge\ \mathbf{F}_{\leqslant 1}\,\bullet\ ) \\ \vee \\ \end{cases}$$

The TPTL formula

$$\mathbf{G}\,({\color{green}\bullet} \rightarrow x.\mathbf{F}\,({\color{red}\bullet} \wedge \mathbf{F}\,({\color{blue}\bullet} \wedge x \leqslant 2)))$$

can be expressed in MTL in the continuous semantics

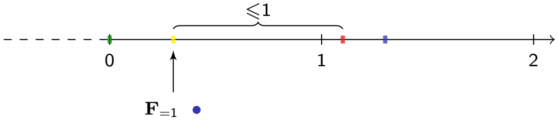


$$\mathbf{G}\ {\color{green}\bullet} \ \rightarrow \ \left\{ \begin{array}{ll} & \mathbf{F}_{\leqslant 1}\ {\color{red}\bullet}\ \wedge\ \mathbf{F}_{[1,2]}\ {\color{blue}\bullet} \\ \vee & \\ & \mathbf{F}_{\leqslant 1}\,(\ {\color{red}\bullet}\ \wedge\ \mathbf{F}_{\leqslant 1}\,{\color{blue}\bullet}\ ) \\ \vee & \\ & \end{array} \right.$$

The TPTL formula

$$\mathbf{G}\left({\color{green}\bullet} \rightarrow x.\mathbf{F}\left({\color{red}\bullet} \wedge \mathbf{F}\left({\color{blue}\bullet} \wedge x \leqslant 2\right)\right)\right)$$

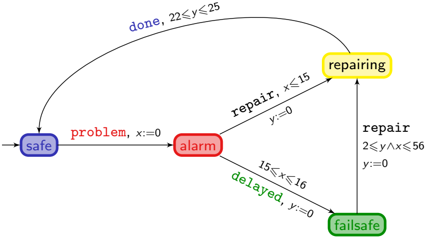
can be expressed in MTL in the continuous semantics



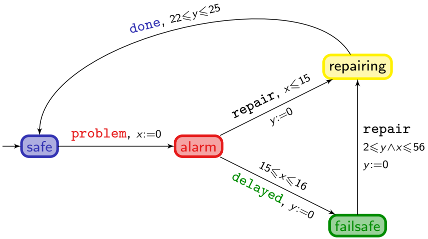
$$\mathbf{G}\ {\color{green}\bullet} \rightarrow \left\{ \begin{array}{l} \mathbf{F}_{\leqslant 1}\ {\color{red}\bullet} \ \wedge\ \mathbf{F}_{[1,2]}\ {\color{blue}\bullet} \\ \vee \\ \mathbf{F}_{\leqslant 1}\left({\color{red}\bullet} \ \wedge\ \mathbf{F}_{\leqslant 1}{\color{blue}\bullet}\right) \\ \vee \\ \mathbf{F}_{\leqslant 1}\ \left(\mathbf{F}_{\leqslant 1}\ {\color{red}\bullet} \ \wedge\ \mathbf{F}_{=1}\ {\color{blue}\bullet}\right) \end{array} \right.$$

Outline

Timed automata

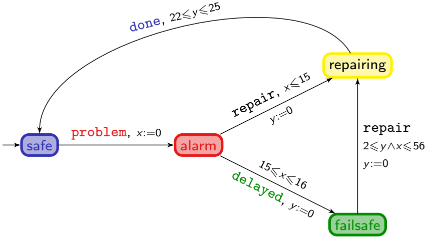


Timed automata



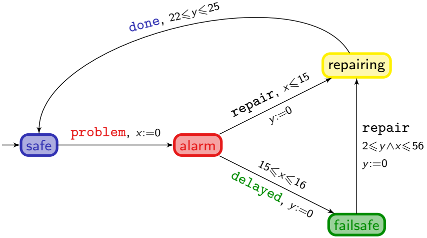
	safe
x	0
y	0

Timed automata



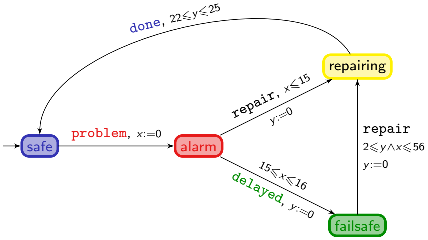
	safe	$\xrightarrow{23}$	safe
x	0		23
y	0		23

Timed automata



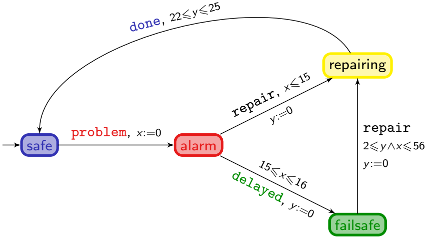
	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm
x	0		23		0
y	0		23		23

Timed automata



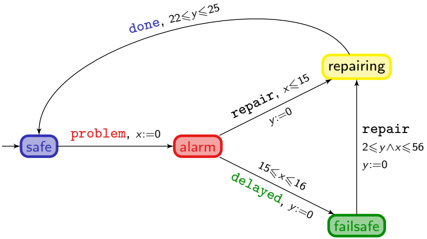
	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm
x	0		23		0		15.6
y	0		23		23		38.6

Timed automata



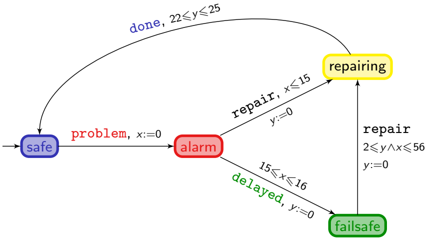
	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\texttt{delayed}}$	failsafe
x	0		23		0		15.6		15.6
y	0		23		23		38.6		0

Timed automata



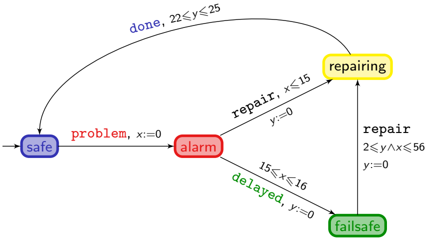
	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\text{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\text{delayed}}$	failsafe	$\xrightarrow{2.3}$	failsafe
x	0		23		0		15.6		15.6		17.9
y	0		23		23		38.6		0		2.3

Timed automata



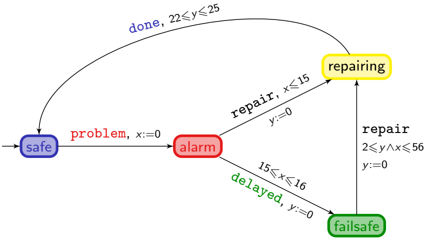
	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\texttt{delayed}}$	failsafe	$\xrightarrow{2.3}$	failsafe	$\xrightarrow{\texttt{repair}}$	reparation
x	0		23		0		15.6		15.6		17.9		17.9
y	0		23		23		38.6		0		2.3		0

Timed automata



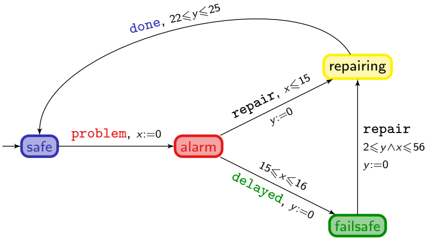
	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\texttt{delayed}}$	failsafe	$\xrightarrow{2.3}$	failsafe	$\xrightarrow{\texttt{repair}}$	reparation	$\xrightarrow{22.1}$	reparation
x	0		23		0		15.6		15.6		17.9		17.9		40
y	0		23		23		38.6		0		2.3		0		22.1

Timed automata



	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\texttt{delayed}}$	failsafe	$\xrightarrow{2.3}$	failsafe	$\xrightarrow{\texttt{repair}}$	reparation	$\xrightarrow{22.1}$	reparation	$\xrightarrow{\texttt{done}}$	safe
x	0		23		0		15.6		15.6		17.9		17.9		40		40
y	0		23		23		38.6		0		2.3		0		22.1		22.1

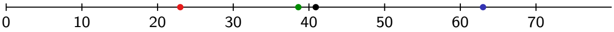
Timed automata



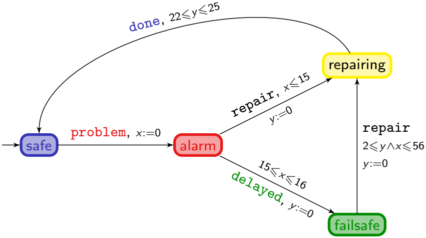
	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\texttt{delayed}}$	failsafe	$\xrightarrow{2.3}$	failsafe	$\xrightarrow{\texttt{repair}}$	reparation	$\xrightarrow{22.1}$	reparation	$\xrightarrow{\texttt{done}}$	safe
x	0		23		0		15.6		15.6		17.9		17.9		40		40
y	0		23		23		38.6		0		2.3		0		22.1		22.1

Can be viewed:

- as the timed word (`problem`,23)(`delayed`,38.6)(`repair`,40.9)(`done`,63)



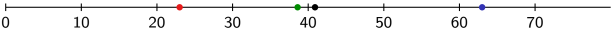
Timed automata



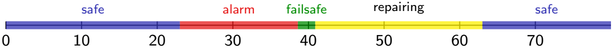
	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\texttt{delayed}}$	failsafe	$\xrightarrow{2.3}$	failsafe	$\xrightarrow{\texttt{repair}}$	reparation	$\xrightarrow{22.1}$	reparation	$\xrightarrow{\texttt{done}}$	safe
x	0		23		0		15.6		15.6		17.9		17.9		40		40
y	0		23		23		38.6		0		2.3		0		22.1		22.1

Can be viewed:

- as the timed word (`problem`,23)(`delayed`,38.6)(`repair`,40.9)(`done`,63)



- as the signal



Basic result on timed automata

Theorem

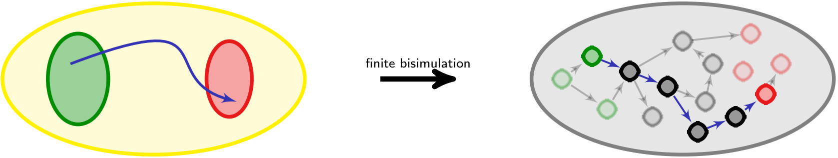
The reachability problem is decidable (and PSPACE-complete) for timed automata [AD94].

[AD94] Alur, Dill. A theory of timed automata (TCS, 1994).

Basic result on timed automata

Theorem

The reachability problem is decidable (and PSPACE-complete) for timed automata [AD94].



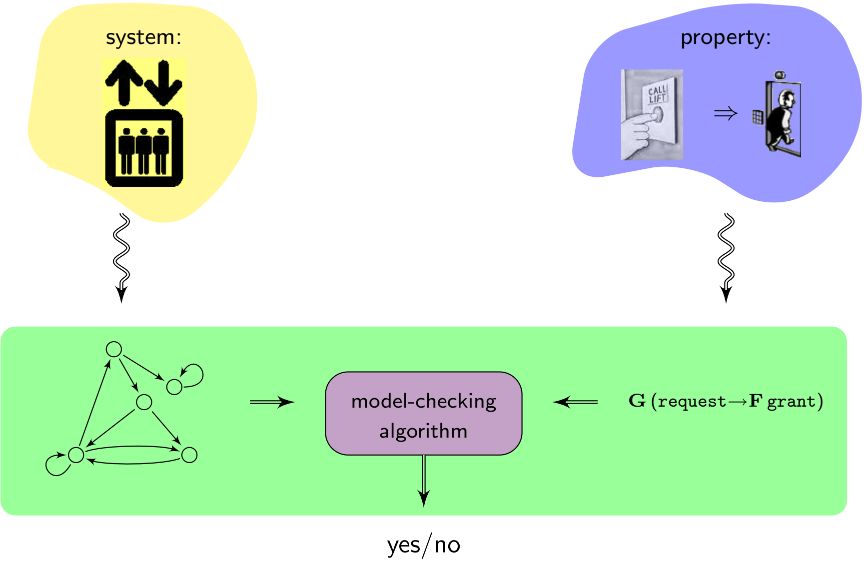
timed automaton

large (but finite) automaton
(region automaton)

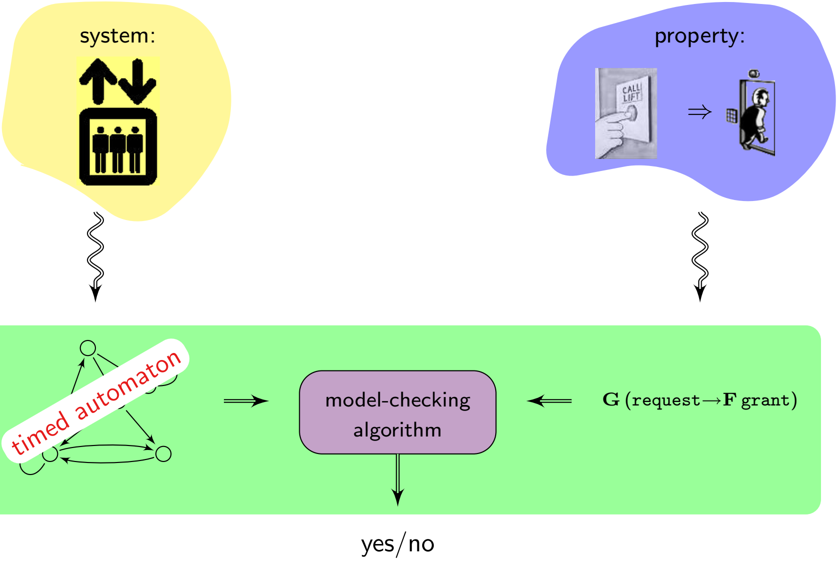
[AD94] Alur, Dill. A theory of timed automata (TCS, 1994).

Outline

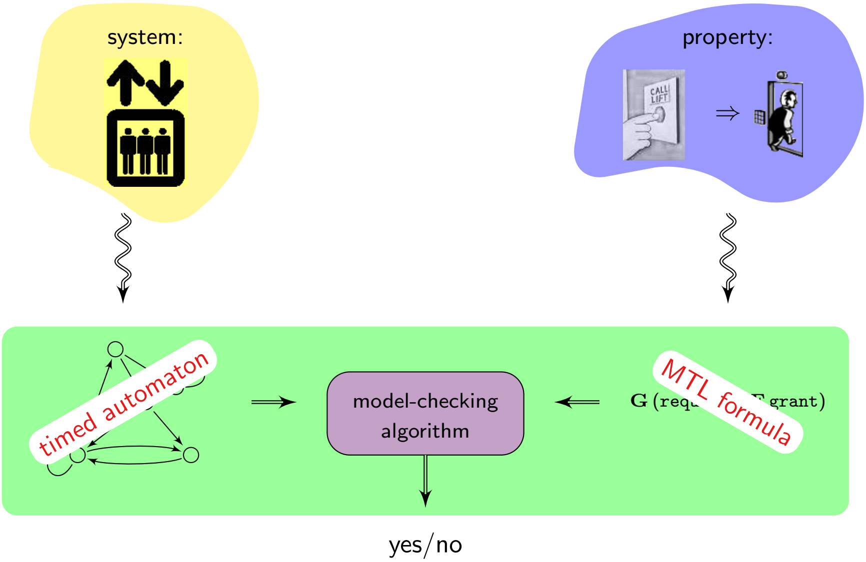
Back to the model-checking problem



Back to the model-checking problem



Back to the model-checking problem



Results

Theorem

Over finite runs, the model-checking problem is:

	pointwise sem.	continuous sem.
MTL	decidable, NPR [OW05]	undecidable [AFH96]
MTL+Past	undecidable	undecidable
TPTL	undecidable [AH94]	undecidable [AH94]

[OW05] Ouaknine, Worrell. On the decidability of metric temporal logic (LICS'05).
[AFH96] Alur, Feder, Henzinger. The benefits of relaxing punctuality (Journal of the ACM, 1996).
[AH94] Alur, Henzinger. A really temporal logic (Journal of the ACM, 1994).

Results

Theorem

Over finite runs, the model-checking problem is:

	pointwise sem.	continuous sem.
MTL	decidable, NPR [OW05]	undecidable [AFH96]
MTL+Past	undecidable	undecidable
TPTL	undecidable [AH94]	undecidable [AH94]

- Model-checking linear-time timed temporal logics is hard!

[OW05] Ouaknine, Worrell. On the decidability of metric temporal logic (LICS'05).
[AFH96] Alur, Feder, Henzinger. The benefits of relaxing punctuality (Journal of the ACM, 1996).
[AH94] Alur, Henzinger. A really temporal logic (Journal of the ACM, 1994).

Results

Theorem

Over finite runs, the model-checking problem is:

	pointwise sem.	continuous sem.
MTL	decidable, NPR [OW05]	undecidable [AFH96]
MTL+Past	undecidable	undecidable
TPTL	undecidable [AH94]	undecidable [AH94]

- Model-checking linear-time timed temporal logics is hard!
- The gap between branching-time and linear-time dramatically increases in the timed framework...

  (reminder: model-checking TCTL is PSPACE-complete)

[OW05] Ouaknine, Worrell. On the decidability of metric temporal logic (LICS'05).
[AFH96] Alur, Feder, Henzinger. The benefits of relaxing punctuality (Journal of the ACM, 1996).
[AH94] Alur, Henzinger. A really temporal logic (Journal of the ACM, 1994).

Results

Theorem

Over finite runs, the model-checking problem is:

	pointwise sem.	continuous sem.
MTL	decidable, NPR [OW05]	undecidable [AFH96]
MTL+Past	undecidable	undecidable
TPTL	undecidable [AH94]	undecidable [AH94]

- Model-checking linear-time timed temporal logics is hard!
- The gap between branching-time and linear-time dramatically increases in the timed framework...

  (reminder: model-checking TCTL is PSPACE-complete)
- All hardness results: by reduction to the halting problem for FIFO channel machines

[OW05] Ouaknine, Worrell. On the decidability of metric temporal logic (LICS'05).
[AFH96] Alur, Feder, Henzinger. The benefits of relaxing punctuality (Journal of the ACM, 1996).
[AH94] Alur, Henzinger. A really temporal logic (Journal of the ACM, 1994).

Results

Theorem

Over finite runs, the model-checking problem is:

	pointwise sem.	continuous sem.
MTL	decidable, NPR [OW05]	undecidable [AFH96]
MTL+Past	undecidable	undecidable
TPTL	undecidable [AH94]	undecidable [AH94]

- Model-checking linear-time timed temporal logics is hard!
- The gap between branching-time and linear-time dramatically increases in the timed framework...

  (reminder: model-checking TCTL is PSPACE-complete)
- All hardness results: by reduction to the halting problem for FIFO channel machines

[OW05] Ouaknine, Worrell. On the decidability of metric temporal logic (LICS'05).
[AFH96] Alur, Feder, Henzinger. The benefits of relaxing punctuality (Journal of the ACM, 1996).
[AH94] Alur, Henzinger. A really temporal logic (Journal of the ACM, 1994).

From LTL to alternating automata

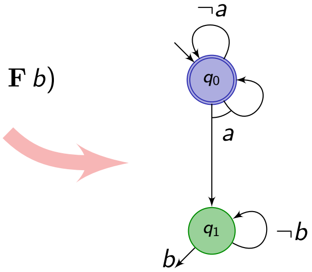
LTL formulas can be turned into linear alternating (Büchi) automata

$$\mathbf{G}\,(a \rightarrow \mathbf{F}\,b)$$

From LTL to alternating automata

LTL formulas can be turned into linear alternating (Büchi) automata

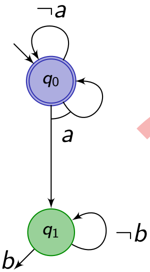
$\mathbf{G}\,(a \rightarrow \mathbf{F}\,b)$

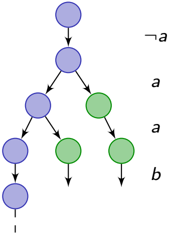


From LTL to alternating automata

LTL formulas can be turned into linear alternating (Büchi) automata

$$\mathbf{G}\,(a \to \mathbf{F}\,b)$$

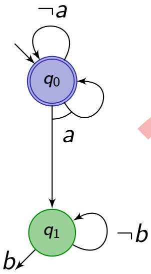


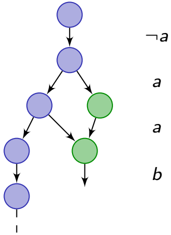


From LTL to alternating automata

LTL formulas can be turned into linear alternating (Büchi) automata

$$\mathbf{G}\,(a \rightarrow \mathbf{F}\,b)$$





From MTL to alternating timed automata

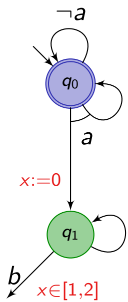
MTL formulas can be turned into linear alternating timed automata

$\mathbf{G}\,(a \rightarrow \mathbf{F}_{[1,2]}\,b)$

From MTL to alternating timed automata

MTL formulas can be turned into linear alternating timed automata

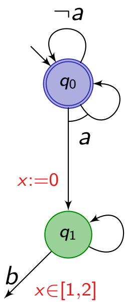
$$\mathbf{G}\,(a \rightarrow \mathbf{F}_{[1,2]}\, b)$$

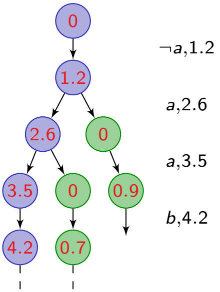


From MTL to alternating timed automata

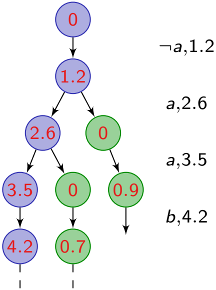
MTL formulas can be turned into linear alternating timed automata

$$\mathbf{G}\,(a \rightarrow \mathbf{F}_{[1,2]}\, b)$$

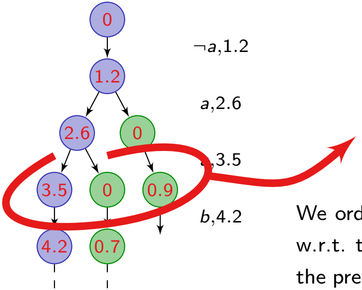


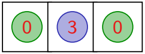


An abstract transition system



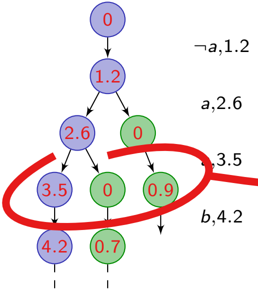
An abstract transition system

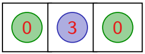




We order elements in a slice of the tree w.r.t. their fractional part, and we forget the precise values of the fractional parts.

An abstract transition system

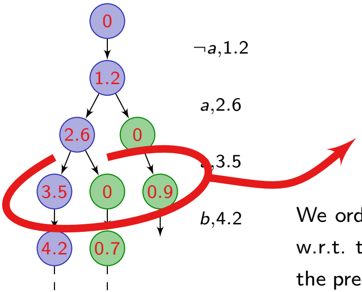


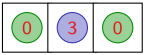


We order elements in a slice of the tree w.r.t. their fractional part, and we forget the precise values of the fractional parts.

☞ this defines an abstract (infinite) transition system

An abstract transition system

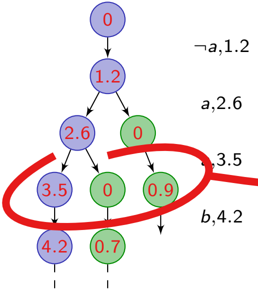


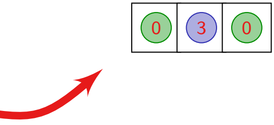


We order elements in a slice of the tree w.r.t. their fractional part, and we forget the precise values of the fractional parts.

- this defines an abstract (infinite) transition system
- it is (time-abstract) bisimilar to the transition system of the alternating timed automata

An abstract transition system





We order elements in a slice of the tree w.r.t. their fractional part, and we forget the precise values of the fractional parts.

- ☞ this defines an abstract (infinite) transition system
- ☞ it is (time-abstract) bisimilar to the transition system of the alternating timed automata
- ☞ there is a well quasi-order on the set of abstract configurations (subword relation):

$$\text{higman} \sqsubseteq \text{highmountain}$$

Summary

Theorem

Over finite runs, the model-checking problem is:

	pointwise sem.	continuous sem.
MTL	decidable, NPR [OW05]	undecidable [AFH96]
MTL+Past	undecidable	undecidable
TPTL	undecidable [AH94]	undecidable [AH94]

What about infinite behaviours?

- the previous algorithm cannot be lifted to the infinite behaviours framework

What about infinite behaviours?

- the previous algorithm cannot be lifted to the infinite behaviours framework
- there is a problem with the accepting condition
  (in the untimed case, we use the Miyano-Hayashi construction [MH84])

What about infinite behaviours?

- the previous algorithm cannot be lifted to the infinite behaviours framework
- there is a problem with the accepting condition
  (in the untimed case, we use the Miyano-Hayashi construction [MH84])

Theorem

Over finite runs, the model-checking problem is:

	pointwise sem.	continuous sem.
MTL	undecidable [OW06]*	undecidable [AFH96]
MTL+Past	undecidable	undecidable
TPTL	undecidable [AH94]	undecidable [AH94]

* by reduction of the recurrence problem for channel machines

[MH84] Miyano, Hayashi. Alternating finite automata on ω-words (TCS, 1984).
[OW06] Ouaknine, Worrell. On metric temporal logic and faulty Turing machines (FoSSaCS'06).

Outline

The fragment without punctuality

- The undecidability/NPR proofs heavily rely on punctual constraints.

The fragment without punctuality

- The undecidability/NPR proofs heavily rely on punctual constraints.

  Old claim: "Any logic strong enough to express the property $\mathbf{G}\,(\textcolor{red}{\bullet} \rightarrow \mathbf{F}_{=1}\,\textcolor{green}{\bullet})$ is undecidable"

The fragment without punctuality

- The undecidability/NPR proofs heavily rely on punctual constraints.

  Old claim: "Any logic strong enough to express the property $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet)$ is undecidable"

- What if we forbid punctual constraints in MTL?

The fragment without punctuality

- The undecidability/NPR proofs heavily rely on punctual constraints.

  Old claim: "Any logic strong enough to express the property $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet)$ is undecidable"
- What if we forbid punctual constraints in MTL?

Metric Interval Temporal Logic (MITL): [AFH96]

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

with I a non-punctual interval

[AFH96] Alur, Feder, Henzinger. The benefits of relaxing punctuality (Journal of the ACM, 1996).

The fragment without punctuality

- The undecidability/NPR proofs heavily rely on punctual constraints.

  Old claim: "Any logic strong enough to express the property $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet)$ is undecidable"
- What if we forbid punctual constraints in MTL?

Metric Interval Temporal Logic (MITL): [AFH96]

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

with I a non-punctual interval

- Examples:
  - $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet)$ is not in MITL

[AFH96] Alur, Feder, Henzinger. The benefits of relaxing punctuality (Journal of the ACM, 1996).

The fragment without punctuality

- The undecidability/NPR proofs heavily rely on punctual constraints.

  Old claim: "Any logic strong enough to express the property $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet)$ is undecidable"
- What if we forbid punctual constraints in MTL?

Metric Interval Temporal Logic (MITL): [AFH96]

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

with I a non-punctual interval

- Examples:
  - $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet)$ is not in MITL
  - $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{[1,2]}\,\bullet)$ is in MITL

[AFH96] Alur, Feder, Henzinger. The benefits of relaxing punctuality (Journal of the ACM, 1996).

Model-checking MITL is “easy”

Theorem

The model-checking problem for MITL is EXPSPACE-complete [AFH96].

Model-checking MITL is “easy”

Theorem

The model-checking problem for MITL is EXPSPACE-complete [AFH96].

☞ we can bound the variability of the signals

Model-checking MITL is "easy"

Theorem

The model-checking problem for MITL is EXPSPACE-complete [AFH96].

- ☞ we can bound the variability of the signals
- ☞ an MITL formula defines a timed regular language

Example: consider the formula $\varphi = \mathbf{G}_{(0,1)}\,(\bullet \rightarrow \mathbf{F}_{[1,2]}\,\bullet)$

Model-checking MITL is "easy"

Theorem

The model-checking problem for MITL is EXPSPACE-complete [AFH96].

- ☞ we can bound the variability of the signals
- ☞ an MITL formula defines a timed regular language

Example: consider the formula $\varphi = \mathbf{G}_{(0,1)} (\bullet \rightarrow \mathbf{F}_{[1,2]} \bullet)$

- each time an $\bullet$ occurs within the first time unit, start a new clock, and check that a $\bullet$ occurs between 1 and 2 time units afterwards

Model-checking MITL is "easy"

Theorem

The model-checking problem for MITL is EXPSPACE-complete [AFH96].

- ☞ we can bound the variability of the signals
- ☞ an MITL formula defines a timed regular language

Example: consider the formula $\varphi = \mathbf{G}_{(0,1)}\,(\bullet \rightarrow \mathbf{F}_{[1,2]}\,\bullet)$

- each time an • occurs within the first time unit, start a new clock, and check that a • occurs between 1 and 2 time units afterwards
- this requires an unbounded number of clocks

Model-checking MITL is "easy"

Theorem

The model-checking problem for MITL is EXPSPACE-complete [AFH96].

- ☞ we can bound the variability of the signals
- ☞ an MITL formula defines a timed regular language

Example: consider the formula $\varphi = \mathbf{G}_{(0,1)}\,(\bullet \rightarrow \mathbf{F}_{[1,2]}\,\bullet)$

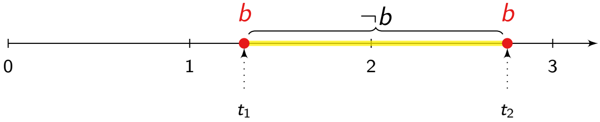
- each time an • occurs within the first time unit, start a new clock, and check that a • occurs between 1 and 2 time units afterwards
- this requires an unbounded number of clocks

☞ something more clever needs to be done

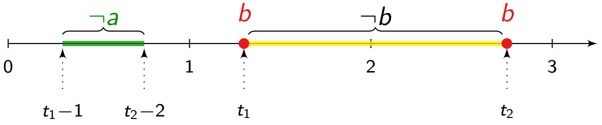
[HR04] Hirshfeld, Rabinovich. Logics for real time: decidability and complexity (Fundamenta Informaticae, 2004).

$$\varphi = \mathbf{G}_{(0,1)}\,(a \rightarrow \mathbf{F}_{[1,2]}\, b)$$

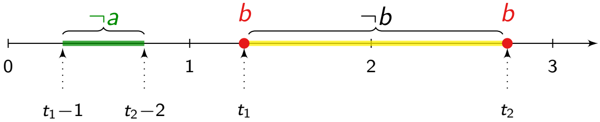
$$\varphi \;=\; \mathbf{G}_{(0,1)}\,(a \rightarrow \mathbf{F}_{[1,2]}\, b)$$

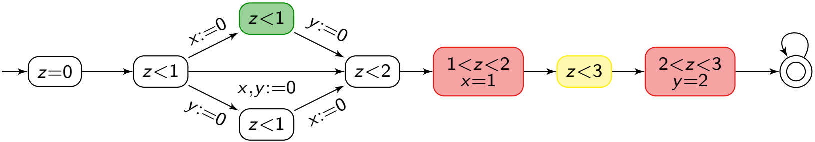


$$\varphi \;=\; \mathbf{G}_{(0,1)}\,(a \rightarrow \mathbf{F}_{[1,2]}\, b)$$

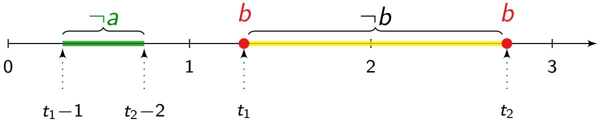


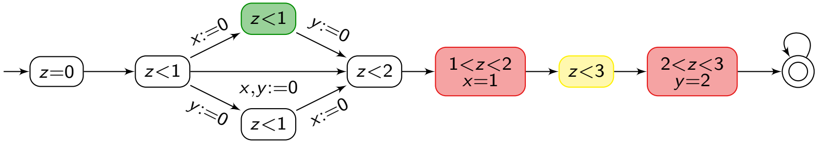
$$\varphi \;=\; \mathbf{G}_{(0,1)}\,(a \rightarrow \mathbf{F}_{[1,2]}\; b)$$





$$\varphi \;=\; \mathbf{G}_{(0,1)}\,(a \rightarrow \mathbf{F}_{[1,2]}\, b)$$





☞ This idea can be extended to any formula in MITL

A co-flat fragment of MTL

- Do punctual constraints really need to be banned?

A co-flat fragment of MTL

- Do punctual constraints really need to be banned?
- Does punctuality always lead to undecidability?

A co-flat fragment of MTL

- Do punctual constraints really need to be banned?
- Does punctuality always lead to undecidability?

We define coFlat-MTL: [BMOW07]

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

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

[BMOW07] Bouyer, Markey, Ouaknine, Worrell. The cost of punctuality (LICS'07).

A co-flat fragment of MTL

- Do punctual constraints really need to be banned?
- Does punctuality always lead to undecidability?

We define coFlat-MTL: [BMOW07]

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

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

- Examples:
  - $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet)$ is in coFlat-MTL

[BMOW07] Bouyer, Markey, Ouaknine, Worrell. The cost of punctuality (LICS'07).

A co-flat fragment of MTL

- Do punctual constraints really need to be banned?
- Does punctuality always lead to undecidability?

We define coFlat-MTL: [BMOW07]

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

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

- Examples:
  - $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet)$ is in coFlat-MTL
  - $\mathbf{F}\,\mathbf{G}_{\leqslant 1}\,\bullet$ is not in coFlat-MTL

[BMOW07] Bouyer, Markey, Ouaknine, Worrell. The cost of punctuality (LICS'07).

A co-flat fragment of MTL

- Do punctual constraints really need to be banned?
- Does punctuality always lead to undecidability?

We define coFlat-MTL: [BMOW07]

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

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

- Examples:
  - $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet)$ is in coFlat-MTL
  - $\mathbf{F}\,\mathbf{G}_{\leqslant 1}\,\bullet$ is not in coFlat-MTL
  - coFlat-MTL contains Bounded-MTL (all modalities are time-bounded)

[BMOW07] Bouyer, Markey, Ouaknine, Worrell. The cost of punctuality (LICS'07).

Model-checking coFlat-MTL is "easy"

Theorem

The model-checking problem for coFlat-MTL or Bounded-MTL is EXPSPACE-complete [BMOW07].

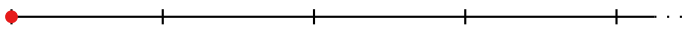
Model-checking coFlat-MTL is "easy"

Theorem

The model-checking problem for coFlat-MTL or Bounded-MTL is EXPSPACE-complete [BMOW07].

- The variability of a Bounded-MTL formula can be high (doubly-exp.):

$$\varphi_n \equiv {\color{red}\bullet} \wedge \mathbf{G}_{[0,2^n]}\, \varphi_D \qquad \text{with} \qquad \varphi_D = \begin{array}{rl} & ({\color{red}\bullet} \rightarrow \mathbf{F}_{=1}\,({\color{red}\bullet} \wedge \mathbf{F}_{\leqslant 1}\,{\color{green}\bullet})) \\ \wedge & ({\color{green}\bullet} \rightarrow \mathbf{F}_{=1}\,({\color{red}\bullet} \wedge \mathbf{F}_{\leqslant 1}\,{\color{green}\bullet})) \end{array}$$



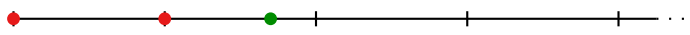
Model-checking coFlat-MTL is "easy"

Theorem

The model-checking problem for coFlat-MTL or Bounded-MTL is EXPSPACE-complete [BMOW07].

- The variability of a Bounded-MTL formula can be high (doubly-exp.):

$$\varphi_n \equiv \textcolor{red}{\bullet} \wedge \mathbf{G}_{[0,2^n]}\,\varphi_D \qquad \text{with} \qquad \varphi_D = \begin{array}{ll} & (\textcolor{red}{\bullet} \to \mathbf{F}_{=1}\,(\textcolor{red}{\bullet} \wedge \mathbf{F}_{\leqslant 1}\,\textcolor{green}{\bullet})) \\ \wedge & (\textcolor{green}{\bullet} \to \mathbf{F}_{=1}\,(\textcolor{red}{\bullet} \wedge \mathbf{F}_{\leqslant 1}\,\textcolor{green}{\bullet})) \end{array}$$



Model-checking coFlat-MTL is “easy”

Theorem

The model-checking problem for coFlat-MTL or Bounded-MTL is EXPSPACE-complete [BMOW07].

- The variability of a Bounded-MTL formula can be high (doubly-exp.):

$$\varphi_n \equiv \bullet \wedge \mathbf{G}_{[0,2^n]}\, \varphi_D \qquad \text{with} \qquad \varphi_D = \begin{array}{ll} & (\bullet \rightarrow \mathbf{F}_{=1}\,(\bullet \wedge \mathbf{F}_{\leqslant 1}\,\bullet)) \\ \wedge & (\bullet \rightarrow \mathbf{F}_{=1}\,(\bullet \wedge \mathbf{F}_{\leqslant 1}\,\bullet)) \end{array}$$

Model-checking coFlat-MTL is "easy"

Theorem

The model-checking problem for coFlat-MTL or Bounded-MTL is EXPSPACE-complete [BMOW07].

- The variability of a Bounded-MTL formula can be high (doubly-exp.):

$$\varphi_n \equiv {\color{red}\bullet} \wedge \mathbf{G}_{[0,2^n]}\, \varphi_D \qquad \text{with} \qquad \varphi_D = \begin{array}{cl} & ({\color{red}\bullet} \rightarrow \mathbf{F}_{=1}\, ({\color{red}\bullet} \wedge \mathbf{F}_{\leqslant 1}\, {\color{green}\bullet})) \\ \wedge & ({\color{green}\bullet} \rightarrow \mathbf{F}_{=1}\, ({\color{red}\bullet} \wedge \mathbf{F}_{\leqslant 1}\, {\color{green}\bullet})) \end{array}$$

Model-checking coFlat-MTL is "easy"

Theorem

The model-checking problem for coFlat-MTL or Bounded-MTL is EXPSPACE-complete [BMOW07].

- The variability of a Bounded-MTL formula can be high (doubly-exp.):

$$\varphi_n \equiv {\color{red}\bullet} \wedge \mathbf{G}_{[0,2^n]}\, \varphi_D \qquad \text{with} \qquad \varphi_D = \begin{array}{rl} & ({\color{red}\bullet} \rightarrow \mathbf{F}_{=1}\,({\color{red}\bullet} \wedge \mathbf{F}_{\leqslant 1}\, {\color{green}\bullet})) \\ \wedge & ({\color{green}\bullet} \rightarrow \mathbf{F}_{=1}\,({\color{red}\bullet} \wedge \mathbf{F}_{\leqslant 1}\, {\color{green}\bullet})) \end{array}$$

Model-checking coFlat-MTL is "easy"

Theorem

The model-checking problem for coFlat-MTL or Bounded-MTL is EXPSPACE-complete [BMOW07].

- The variability of a Bounded-MTL formula can be high (doubly-exp.):

$$\varphi_n \equiv \textcolor{red}{\bullet} \wedge \mathbf{G}_{[0,2^n]}\, \varphi_D \qquad \text{with} \qquad \varphi_D = \begin{array}{rl} & (\textcolor{red}{\bullet} \rightarrow \mathbf{F}_{=1}\,(\textcolor{red}{\bullet} \wedge \mathbf{F}_{\leqslant 1}\,\textcolor{green}{\bullet})) \\ \wedge & (\textcolor{green}{\bullet} \rightarrow \mathbf{F}_{=1}\,(\textcolor{red}{\bullet} \wedge \mathbf{F}_{\leqslant 1}\,\textcolor{green}{\bullet})) \end{array}$$

- A Bounded-MTL formula may define a non timed-regular language:

$$\mathbf{G}_{\leqslant 1}\,(\textcolor{red}{\bullet} \rightarrow \mathbf{F}_{=1}\,\textcolor{green}{\bullet}) \wedge \mathbf{G}_{\leqslant 1}\,\textcolor{red}{\bullet} \wedge \mathbf{G}_{(1,2]}\,\textcolor{green}{\bullet}$$

defines the context-free language $\{\textcolor{red}{\bullet}^n \textcolor{green}{\bullet}^m \mid n \leqslant m\}$.

Algorithm for Bounded-MTL

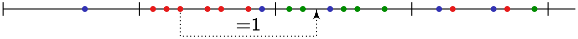
Assume one wants to verify formula

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

Algorithm for Bounded-MTL

Assume one wants to verify formula

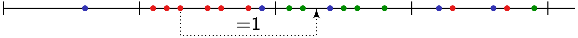
$$\mathbf{G}_{<2}\left({\color{red}\bullet} \rightarrow \mathbf{F}_{=1}\,{\color{green}\bullet}\right)$$



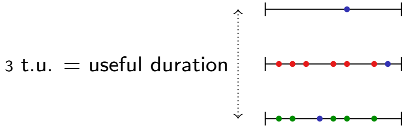
Algorithm for Bounded-MTL

Assume one wants to verify formula

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



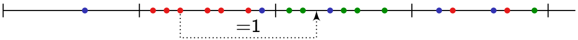
Offline, we stack all 'relevant' time units and use a sliding window:



Algorithm for Bounded-MTL

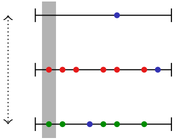
Assume one wants to verify formula

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



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

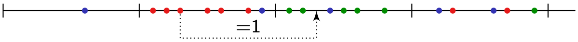
3 t.u. = useful duration



Algorithm for Bounded-MTL

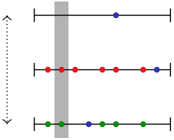
Assume one wants to verify formula

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



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

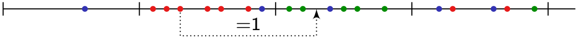
3 t.u. = useful duration



Algorithm for Bounded-MTL

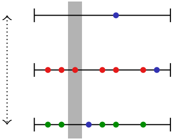
Assume one wants to verify formula

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



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

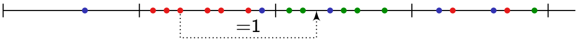
3 t.u. = useful duration



Algorithm for Bounded-MTL

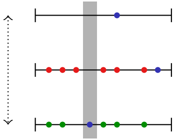
Assume one wants to verify formula

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



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

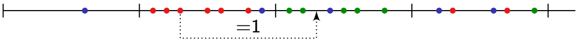
3 t.u. $=$ useful duration



Algorithm for Bounded-MTL

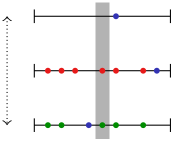
Assume one wants to verify formula

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



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

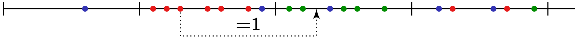
3 t.u. = useful duration



Algorithm for Bounded-MTL

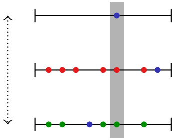
Assume one wants to verify formula

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



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

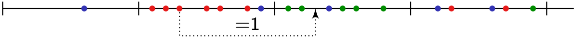
3 t.u. = useful duration



Algorithm for Bounded-MTL

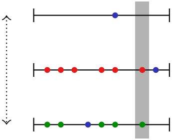
Assume one wants to verify formula

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



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

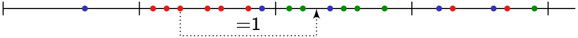
3 t.u. = useful duration



Algorithm for Bounded-MTL

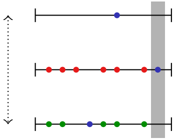
Assume one wants to verify formula

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



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

3 t.u. = useful duration

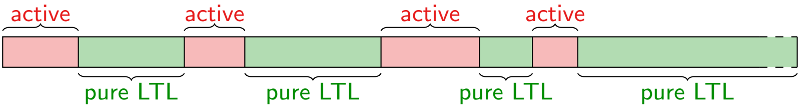


Algorithm for coFlat-MTL

$\varphi \rightsquigarrow$ alternating timed automata $\mathcal{B}_{\neg\varphi}$ for $\neg\varphi$ with a 'flatness' property

Algorithm for coFlat-MTL

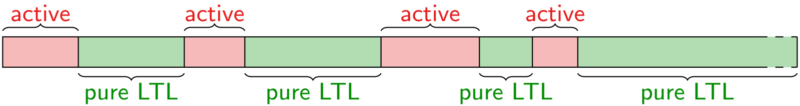
$\varphi \rightsquigarrow$ alternating timed automata $\mathcal{B}_{\neg\varphi}$ for $\neg\varphi$ with a 'flatness' property



where - the number of active fragments is at most exponential
- the total duration of active fragments is at most exponential

Algorithm for coFlat-MTL

$\varphi \rightsquigarrow$ alternating timed automata $\mathcal{B}_{\neg\varphi}$ for $\neg\varphi$ with a 'flatness' property



where - the number of active fragments is at most exponential
- the total duration of active fragments is at most exponential

- active fragment = cycle-bounded computation in a channel machine
- pure LTL part = finite automaton computation

Outline

Conclusion

- Recent advances have raised a new interest for linear-time timed temporal logics
  - Not everything is undecidable
  - Some rather ‘efficient’ subclasses
    - non-punctual formulas
    - structurally (co-)flat formulas

Conclusion

- Recent advances have raised a new interest for linear-time timed temporal logics
  - Not everything is undecidable
  - Some rather ‘efficient’ subclasses
    - non-punctual formulas
    - structurally (co-)flat formulas
- A recent result: coFlat-MTL$_{\text{MITL}}$ unifies coFlat-MTL and MITL, and is EXPSPACE-complete [BMOW08]!

Conclusion

- Recent advances have raised a new interest for linear-time timed temporal logics
  - Not everything is undecidable
  - Some rather 'efficient' subclasses
    - non-punctual formulas
    - structurally (co-)flat formulas
- A recent result: coFlat-$\mathsf{MTL}_{\mathsf{MITL}}$ unifies coFlat-MTL and MITL, and is EXPSPACE-complete [BMOW08]!
- No real data structures do exist for these logics.

[BMOW08] Bouyer, Markey, Ouaknine, Worrell. On Expressiveness and Complexity in Real-time Model Checking. Submitted.