Logiques pour les systèmes temporisés : contrôle et expressivité

Fabrice Chevalier

LSV - ENS Cachan & CNRS

25 juin 2007
Systems including a software component
Usual approach

Most used validation approach: Test

but it is not exhaustive ···
Usual approach

Most used validation approach:

Test

but it is not exhaustive …
Why verify systems formally?

Error Knocks Down Google $350 a Share

By DAN DORFMAN
July 31, 2006

It was like a Wall Street version of one of those Keystone Cops comedies of the 1930s, but at least one investor wasn't laughing. According to his stockbroker, he almost had a heart attack as a result of what appeared to be an enormous plunge Thursday in after-hours trading in the price of the high-flying shares of Internet search engine Google.
We want to **prove** that systems behave correctly

**Formal methods:**

- Model-based testing
- Static analysis
- Theorem proving
- Model-checking
We want to **prove** that systems behave correctly

**Formal methods:**
- Model-based testing
- Static analysis
- Theorem proving
- Model-checking
Model-checking

Does the system satisfy the property?

Modelling

\( \varphi \)
Model-checking

Does the system satisfy the property?

Model-checking algorithm

Modelling
Controller synthesis

Can we guide the system so that it satisfies the property?

Modelling

\[ \varphi \]
Controller synthesis

Can we guide the system so that it satisfies the property?

Modelling
Real systems naturally use **timing**:

- The doors of the subway must stay open between 20 and 30 seconds.
- The car must reach the required speed **within** one minute.

Models of systems must involve **time**

→ **Timed control.**
Control problem

Controllable and uncontrollable actions

2 players
Control problem

Controllable and uncontrollable actions

2 players
- The controller
Control problem

Controllable and uncontrollable actions

2 players
- The controller
- The environnement
Control problem

Controllable and uncontrollable actions

2 players

- The controller

- The environnement

A winning condition
Control: finite case
Control: finite case

propose $c_1$
Control: finite case
Control: finite case

propose $e_1$
Control: finite case
Control: finite case

propose $c_1$
Control: finite case
Control: finite case

propose $e_1$
Control: finite case
Control: finite case

Second try
Control: finite case

propose $c_2$
Control: finite case
Control: finite case
Control: finite case
Control: finite case

propose $c_1$
Control: finite case
Outline

1 Verification and control of o-minimal systems
   - Motivations
   - Control of o-minimal systems
   - Weighted o-minimal hybrid systems

2 Expressivity and control for MTL
   - Real time logics
   - Expressivity of MTL and TPTL
   - Control for MTL specifications

3 Automata representation for timed logics
   - Existing results
   - Input-determined automata and logics
   - Expressiveness results
Outline

1 Verification and control of o-minimal systems
   - Motivations
     - Control of o-minimal systems
     - Weighted o-minimal hybrid systems

2 Expressivity and control for MTL
   - Real time logics
   - Expressivity of MTL and TPTL
   - Control for MTL specifications

3 Automata representation for timed logics
   - Existing results
   - Input-determined automata and logics
   - Expressiveness results
Controller synthesis

Can we guide the system so that it satisfies the property?

Modelling
A winter example

Room

Heater
A winter example
A winter example

- Room
- Thermostat
- Heater

Max

min
A winter example
A winter example

- Thermostat
- Room
- Heater
- Cold Air
A winter example - natural question

Whatever the environment does, can we ensure that the temperature remains between min and Max?
A winter control problem

2 players:
- The controller can switch on/off the heater
- The environment can open/close the door

Winning condition:
the temperature remains between min and Max
A winter control problem

Heater OFF

\[ c, X = \text{min} \]

Door Closed

e, e

Door Open

Heater OFF

\[ c, X = \text{min} \]

Heater ON

\[ c, X = \text{Max} \]

Door Closed

e, e

Door Open

Heater ON

\[ c, X = \text{Max} \]
A winter control problem

\[ \dot{X} = -KX \]

- Heater OFF
- Door Closed

\[ \dot{X} = -K(d + X) \]

- Heater OFF
- Door Open

\[ \dot{X} = K(h - d - X) \]

- Heater ON
- Door Open

\[ \dot{X} = K(h - X) \]

- Heater ON
- Door Closed

\[ c, \ X = \text{min} \]

\[ c, \ X = \text{Max} \]
A winter example: temperature curves

\[ \theta \]

\[ \theta_0 \]

Door Closed
Heater On

Door Open
Heater On

Door Closed
Heater Off

Door Open
Heater Off

\[ t \]
Outline

1. Verification and control of o-minimal systems
   - Motivations
   - Control of o-minimal systems
   - Weighted o-minimal hybrid systems

2. Expressivity and control for MTL
   - Real time logics
   - Expressivity of MTL and TPTL
   - Control for MTL specifications

3. Automata representation for timed logics
   - Existing results
   - Input-determined automata and logics
   - Expressiveness results
Hybrid systems

Finite automaton

[Diagram of a finite automaton with three states and transitions between them]
Hybrid systems

Finite automaton
+ continuous variables

[Diagram of a finite automaton connected to two other elements]
Hybrid systems

Finite automaton

+ continuous variables
+ guards and resets

\[ \theta = \max, c, \theta := \theta_1 \]
O-minimal Structure

**O-minimal Structure** [PS86] [vdD96]

\[ \mathcal{M} = \langle M, <, \cdots \rangle \] is an *o-minimal structure* if every first-order definable subset of \( M \) is a finite union of points and open intervals (possibly unbounded).

**Examples of o-minimal structures**

- \( \langle \mathbb{Q}, \leq \rangle, \langle \mathbb{Q}, \leq, +, 0 \rangle \),
- \( \langle \mathbb{R}, \leq, +, \cdot, 0, 1 \rangle \),
- \( \langle \mathbb{R}, \leq, +, \cdot, 0, 1, e^x \rangle \) [Wi96].
O-minimal Hybrid System

O-minimal hybrid system [LPS00]

An **o-minimal hybrid system** is a hybrid system where

- The dynamic, guards and resets are definable in an o-minimal structure,
- At each discrete step, every variable is reset independently from its initial value.

Rich dynamics but **strong reset condition**
Example of o-minimal Dynamical Systems

Timed dynamics

\[ \gamma : (\mathbb{R}^+)^2 \times \mathbb{R}^+ \rightarrow (\mathbb{R}^+)^2 \]

\[ \gamma(x_1, x_2, t) = (x_1 + t, x_2 + t) \]
Example of o-minimal Dynamical Systems

Rectangular dynamics

\[ \gamma : (\mathbb{R}^+)^2 \times [1, 2] \times \mathbb{R}^+ \rightarrow (\mathbb{R}^+)^2 \]

\[ \gamma(x_1, x_2, p, t) = (x_1 + t, x_2 + p \cdot t) \]
A winter control problem

**Heater OFF**

- $\dot{X} = -KX$
- $\theta = \theta_0 e^{-K \cdot t}$

**Heater ON**

- $\dot{X} = K(h - X)$
- $\theta = e^{-K \cdot t}(\theta_0 - h) + h$

**Door Closed**

- $c, X = \text{min}$
- $c, X = \text{Max}$

**Door Open**

- $\dot{X} = -K(d + X)$
- $\theta = e^{-K \cdot t}(\theta_0 + d) - d$

- $\dot{X} = K(h - d - X)$
- $\theta = e^{-K \cdot t}(\theta_0 - h + d) + h - d$

- $c, X = \text{min}$
- $c, X = \text{Max}$
Theorem (LICS’06)

Let $\mathcal{C}$ be a class of o-minimal hybrid systems defined on $\mathcal{M}$ such that $\text{Th}(\mathcal{M})$ is decidable, then

- the control problem in $\mathcal{C}$ is decidable.

Indeed, given an o-minimal hybrid system $\mathcal{H} \in \mathcal{C}$

- the set of winning states is computable,
- we can give a strategy definable in $\mathcal{M}$ for each winning states.
Results on control of o-minimal hybrid systems

Theorem (LICS’06)

Let $\mathcal{C}$ be a class of o-minimal hybrid systems defined on $\mathcal{M}$ such that $Th(\mathcal{M})$ is decidable, then

- the control problem in $\mathcal{C}$ is decidable.

Indeed, given an o-minimal hybrid system $\mathcal{H} \in \mathcal{C}$

- the set of winning states is computable,
- we can give a strategy definable in $\mathcal{M}$ for each winning states.
Results on control of o-minimal hybrid systems

Theorem (LICS'06)

Let $\mathcal{C}$ be a class of o-minimal hybrid systems defined on $\mathcal{M}$ such that $Th(\mathcal{M})$ is decidable, then

- the control problem in $\mathcal{C}$ is decidable.

Indeed, given an o-minimal hybrid system $\mathcal{H} \in \mathcal{C}$

- the set of winning states is computable,
- we can give a strategy definable in $\mathcal{M}$ for each winning states.
Motivations
Control of o-minimal systems
Weighted o-minimal hybrid systems

Illustration on THE spiral

Goal

$q_0$

$g_c$, $c$

$g_e$, $e$
Motivations
Control of o-minimal systems
Automata representation for timed logics

Illustration on THE spiral

$q_0 \rightarrow \text{Goal}$

$g_c, c$

$g_e, e$

Dynamic in $q_0$
Illustration on THE spiral

- The guard $g_e$
- $g_c, c$
- $g_e, e$
- $q_0$ to Goal
Illustration on THE spiral

The guard $g_c$

$q_0$ → Goal

$g_c, c$

$g_e, e$
Illustration on THE spiral

Goal

$q_0$

$g_c, c$

$g_e, e$

Winning strategy from $(0, 0)$
Outline

1. Verification and control of o-minimal systems
   - Motivations
   - Control of o-minimal systems
   - Weighted o-minimal hybrid systems

2. Expressivity and control for MTL
   - Real time logics
   - Expressivity of MTL and TPTL
   - Control for MTL specifications

3. Automata representation for timed logics
   - Existing results
   - Input-determined automata and logics
   - Expressiveness results
A winter example
A winter example - new natural question

How much energy is needed to reach the Max temperature, whatever the environment does?
Weighted o-minimal hybrid system

We add Cost functions

\[ \text{Cost} = \tau \]

\[ \text{Cost} = 5.\tau^2 \]

\[ \text{Cost} = e^\tau \]
Results on weighted o-minimal hybrid system

Goal

- Determine if it is possible for the controller to reach a winning state whatever the environment does with a given cost.
- Compute the optimal cost to reach winning states whatever the environment does.

Theorem (LFCS'07)

Let $\mathcal{M}$ be an o-minimal structure such that $Th(\mathcal{M})$ is decidable, then:

- The cost-optimal reachability control problem is decidable.
- The optimal cost is computable.
Results on weighted o-minimal hybrid system

Goal
- Determine if it is possible for the controller to reach a winning state whatever the environment does with a given cost.
- Compute the optimal cost to reach winning states whatever the environment does.

Theorem (LFCS’07)
Let $\mathcal{M}$ be an o-minimal structure such that $Th(\mathcal{M})$ is decidable, then:
- The cost-optimal reachability control problem is decidable.
- The optimal cost is computable.
Outline

1 Verification and control of o-minimal systems
   - Motivations
   - Control of o-minimal systems
   - Weighted o-minimal hybrid systems

2 Expressivity and control for MTL
   - Real time logics
   - Expressivity of MTL and TPTL
   - Control for MTL specifications

3 Automata representation for timed logics
   - Existing results
   - Input-determined automata and logics
   - Expressiveness results
Controller synthesis

Can we guide the system so that it satisfies the property?

Modelling
Linear time temporal logic

Syntax of LTL ([Pnu77]):

\[
\text{LTL} \ni \varphi ::= p \mid \neg \varphi \mid \varphi \lor \varphi \mid \varphi \mathbf{U} \varphi
\]

Example: After each problem there is an alarm.

\[
\begin{array}{cccc}
\text{start} & \text{problem} & \text{action} & \text{alarm} \\
\hline
\end{array}
\quad \models \text{G} (\text{problem} \Rightarrow \text{F alarm})
\]
Adding real-time constraints

Syntax of MTL ([Koy90]):

\[ \text{MTL} \ni \varphi ::= p \mid \neg \varphi \mid \varphi \lor \varphi \mid \varphi \mathsf{U} t, \varphi \]

After each problem there is an alarm in less than 5 time units.

![Diagram 1](start 0 -- problem 5.4 -- alarm 9.1 = G (problem \Rightarrow F_{\leq 5} \text{alarm}))

![Diagram 2](start 0 -- problem 3 -- alarm 9.1 \neq G (problem \Rightarrow F_{\leq 5} \text{alarm}))
Adding real-time constraints

Syntax of MTL ([Koy90]):

\[ \text{MTL } \ni \varphi ::= p \mid \neg \varphi \mid \varphi \lor \varphi \mid \varphi U I \varphi \]

After each \textbf{problem} there is an \textbf{alarm} in less than 5 time units.

\[ \begin{array}{cccc}
\text{start} & \text{problem} & \text{alarm} & \models \quad \text{G} (\text{problem } \Rightarrow \text{ F } \leq 5 \text{alarm})
\end{array} \]

\[ \begin{array}{cccc}
\text{start} & \text{problem} & \text{alarm} & \not\models \quad \text{G} (\text{problem } \Rightarrow \text{ F } \leq 5 \text{alarm})
\end{array} \]
Adding real-time constraints

Syntax of MTL ([Koy90]):

\[ MTL \ni \varphi ::= p \mid \neg \varphi \mid \varphi \lor \varphi \mid \varphi \mathbf{U} \varphi \]

After each **problem** there is an **alarm** in less than 5 time units.

\[
\begin{align*}
\text{start} & \quad \text{problem} & \quad \text{alarm} & \quad \models G (\text{problem} \Rightarrow F_{\leq 5} \text{alarm}) \\
0 & \quad 5.4 & \quad 9.1
\end{align*}
\]

\[
\begin{align*}
\text{start} & \quad \text{problem} & \quad \text{alarm} & \quad \not\models G (\text{problem} \Rightarrow F_{\leq 5} \text{alarm}) \\
0 & \quad 3 & \quad 9.1
\end{align*}
\]
Adding real-time constraints

Syntax of TPTL ([AH89]):

\[
\text{TPTL } \ni \varphi ::= p \mid x \sim c \mid \neg \varphi \mid \varphi \lor \varphi \mid \varphi \mathbf{U} \varphi \mid x. \varphi
\]

After each problem there is an alarm in less than 5 time units.

\[
G(\text{problem } \Rightarrow x. \mathbf{F}(\text{alarm} \land x \leq 5))
\]

After each problem there is an alarm, and then a reboot in less than 2 time units.

\[
G(\text{problem } \Rightarrow x. \mathbf{F}(\text{alarm} \land \mathbf{F}(\text{reboot} \land x \leq 2)))
\]
Adding real-time constraints

Syntax of TPTL ([AH89]):

\[
\text{TPTL } \ni \phi ::= p \mid x \sim c \mid \neg \phi \mid \phi \lor \phi \mid \phi \ U \phi \mid x. \phi
\]

After each problem there is an alarm in less than 5 time units.

\[
G(\text{problem } \Rightarrow x. F(\text{alarm } \land x \leq 5))
\]

After each problem there is an alarm, and then a reboot in less than 2 time units.

\[
G(\text{problem } \Rightarrow x. F(\text{alarm } \land F(\text{reboot } \land x \leq 2)))
\]
Adding real-time constraints

Syntax of TPTL ([AH89]):

\[
\text{TPTL} \ni \varphi ::= p \mid x \sim c \mid \neg \varphi \mid \varphi \lor \varphi \mid \varphi \cup \varphi \mid x. \varphi
\]

After each problem there is an alarm in less than 5 time units.

\[
G(\text{problem} \Rightarrow x.F(\text{alarm} \land x \leq 5))
\]

After each problem there is an alarm, and then a reboot in less than 2 time units.

\[
G(\text{problem} \Rightarrow x.F(\text{alarm} \land F(\text{reboot} \land x \leq 2)))
\]
Outline

1. Verification and control of o-minimal systems
   - Motivations
   - Control of o-minimal systems
   - Weighted o-minimal hybrid systems

2. Expressivity and control for MTL
   - Real time logics
   - Expressivity of MTL and TPTL
   - Control for MTL specifications

3. Automata representation for timed logics
   - Existing results
   - Input-determined automata and logics
   - Expressiveness results
Relative expressiveness of MTL and TPTL

Clearly, MTL can be translated into TPTL:

\[ \varphi U_{\sim c} \psi \equiv x. \varphi U (\psi \land x \sim c). \]

\[ \cdots \text{ but does the converse hold?} \]

**Conjecture [AH90]**

- TPTL is strictly more expressive than MTL,
- the TPTL formula

\[ G (a \Rightarrow x. F (b \land F (c \land x \leq 2))) \]

cannot be expressed in MTL.
Clearly, MTL can be translated into TPTL:
\[ \varphi \bigcup_{x} \psi \equiv x. \varphi \bigcup (\psi \land x \sim c). \]

\[ \cdots \text{but does the converse hold?} \]

**Conjecture [AH90]**

- **TPTL** is strictly more expressive than MTL,
- the **TPTL** formula

\[ G (a \Rightarrow x. F (b \land F (c \land x \leq 2))) \]

cannot be expressed in **MTL**.
The formula can be expressed...

It turns out that the formula can be expressed in MTL:

$$G (a \implies x. F (b \land F (c \land x \leq 2)))$$
It turns out that the formula can be expressed in MTL:

\[ G (a \Rightarrow x. \ F (b \land \ F (c \land x \leq 2))) \]
The formula can be expressed...

It turns out that the formula can be expressed in MTL:

\[ G(a \Rightarrow x. F(b \land F(c \land x \leq 2))) \]
The formula can be expressed...

It turns out that the formula can be expressed in MTL:

\[ G (a \Rightarrow x. \, F (b \land F (c \land x \leq 2))) \]

\[ G \ a \Rightarrow \begin{cases} \bigvee \left( F_{\leq 1} b \land F_{[1,2]} c \right) \\ \bigvee \left( F_{\leq 1} (b \land F_{\leq 1} c) \right) \\ \bigvee \left( F_{\leq 1} (F_{\leq 1} b \land F_{=1} c) \right) \end{cases} \]
The formula can be expressed...

It turns out that the formula can be expressed in MTL:

$$G (a \Rightarrow (b \land F(c \land x \leq 2)))$$

$$G (a \Rightarrow \bigvee \{ F_{\leq 1} b \land F_{[1,2]} c \}$$

$$\bigvee F_{\leq 1} (b \land F_{\leq 1} c)$$

$$\bigvee F_{\leq 1} (F_{\leq 1} b \land F_{=1} c)$$
The formula can be expressed...

It turns out that the formula can be expressed in MTL:

\[ G(a \Rightarrow x. F(b \land F(c \land x \leq 2))) \]

\[
\begin{cases}
    F_{\leq 1} b \land F_{[1,2]} c \\
    \lor \\
    F_{\leq 1} (b \land F_{\leq 1} c) \\
    \lor \\
    F_{\leq 1} (F_{\leq 1} b \land F_{=1} c)
\end{cases}
\]
The formula can be expressed...

It turns out that the formula can be expressed in MTL:

\[
G(a \Rightarrow x. F(b \land F(c \land x \leq 2)))
\]

\[
F_{\leq 1} b \land F_{[1,2]} c
\]

\[
G a \Rightarrow \left\{
\begin{array}{l}
F_{\leq 1} (b \land F_{\leq 1} c) \\
\lor \\
F_{\leq 1} (F_{\leq 1} b \land F_{= 1} c)
\end{array}
\right.
\]
The formula can be expressed...

\[ x \cdot F(b \land F(c \land x \leq 2)) \equiv \begin{cases} 
F_{\leq 1} b \land F_{=1} F_{\leq 1} c \\
\lor \\
F_{\leq 1} (b \land F_{\leq 1} c) \\
\lor \\
F_{\leq 1} (F_{\leq 1} b \land F_{=1} c) 
\end{cases} \]

- it does not mean that TPTL \(\equiv\) MTL.
- the formulas are equivalent only for the continuous semantics.
The formula can be expressed...

\[ x \cdot F \left( b \land F \left( c \land x \leq 2 \right) \right) \equiv \begin{cases} \ldots \equiv \begin{cases} F \leq_1 b \land F =_1 F \leq_1 c \\ \lor \\ F \leq_1 (b \land F \leq_1 c) \\ \lor \\ F \leq_1 (F \leq_1 b \land F =_1 c) \end{cases} \end{cases} \]

- It does not mean that \( \text{TPTL} \equiv \text{MTL} \).
- The formulas are equivalent only for the continuous semantics.
... but the result still holds!

Theorem (FSTTCS’05)

- **TPTL** is strictly more expressive than **MTL** for *both* semantics,
- the **TPTL** formula

$$G(a \Rightarrow x. F(b \land F(c \land x \leq 2)))$$

cannot be expressed in **MTL** under the pointwise semantics,
- the **TPTL** formula

$$x. F(a \land x \leq 1 \land G(x \leq 1 \Rightarrow \neg b))$$

cannot be expressed in **MTL** under both semantics.
Outline

1 Verification and control of o-minimal systems
   • Motivations
   • Control of o-minimal systems
   • Weighted o-minimal hybrid systems

2 Expressivity and control for MTL
   • Real time logics
   • Expressivity of MTL and TPTL
   • Control for MTL specifications

3 Automata representation for timed logics
   • Existing results
   • Input-determined automata and logics
   • Expressiveness results
Controller synthesis

Can we guide the system so that it satisfies the property?

Modelling

verification and control of o-minimal systems
expressivity and control for MTL
automata representation for timed logics
real time logics
expressivity of MTL and TPTL
control for MTL specifications
The MTL control problem for timed automata

A system to control:

\[ x \geq 2, \text{alarm}, x := 0 \]

\[ y \geq 6, \text{alarm} \]

A property to satisfy:

\[ G(\text{problem} \Rightarrow F_{\leq 5}\text{alarm}) \]
The MTL control problem for timed automata

A system to control:

\[ x \geq 2, \text{alarm}, x := 0 \]

\[ y \geq 6, \text{alarm} \]

A property to satisfy:

\[ G(\text{problem} \Rightarrow F_{\leq 5}\text{alarm}) \]
The MTL control problem for timed automata

Is there a controller given as a timed automaton such that whatever the environment does the property is satisfied?

Theorem (CONCUR’06)
The control problem for timed automata for MTL specifications over finite words is undecidable.
Control with bounded resources

∃?

is complicated
Control with bounded resources

∃? is complicated

Fixing the resources of the controller.
→ Example: 7 clocks, integer constants smaller than 5

∃? is easier
Fixing the resources of the controller.

Example: 7 clocks, integer constants smaller than 5

∃? is easier

Theorem (CONCUR'06)

The control problem for timed automata for MTL specifications with bounded resources over finite words is decidable.
Outline

1 Verification and control of o-minimal systems
   - Motivations
   - Control of o-minimal systems
   - Weighted o-minimal hybrid systems

2 Expressivity and control for MTL
   - Real time logics
   - Expressivity of MTL and TPTL
   - Control for MTL specifications

3 Automata representation for timed logics
   - Existing results
   - Input-determined automata and logics
   - Expressiveness results
Can we guide the system so that it satisfies the property?

Modelling
MSO and finite automata

Büchi’s theorem:

\[
\text{MSO: } \varphi ::= Q_a(x) \mid x < y \mid x \in X \mid \neg \varphi \mid \varphi \lor \varphi \mid \exists x \varphi \mid \exists X \varphi
\]

MSO-formulas $\iff$ Finite automata

\[
\exists X (0 \in X \land (\forall x \in X \ Q_a(x)) \land (\forall x \in X \Rightarrow x + 2 \in X))
\]
More general picture

MSO .......................................................... FA

Expressiveness results

Input-determined automata and logics

Existing results

Automata representation for timed logics

Verification and control of o-minimal systems

Expressivity and control for MTL
More general picture
More general picture

- MSO
- FO
- LTL
- FA

Existing results
Input-determined automata and logics
Expressiveness results

Verification and control of o-minimal systems
Expressivity and control for MTL
Automata representation for timed logics
More general picture
More general picture

Goal: same framework for timed logics and automata under the continuous semantics
Outline

1. Verification and control of o-minimal systems
   - Motivations
   - Control of o-minimal systems
   - Weighted o-minimal hybrid systems

2. Expressivity and control for MTL
   - Real time logics
   - Expressivity of MTL and TPTL
   - Control for MTL specifications

3. Automata representation for timed logics
   - Existing results
   - Input-determined automata and logics
   - Expressiveness results
Input-determined operators

Generic class of timed operators $\Delta$

Example of the operator $\diamond_a$

$\diamond^l_a(\sigma, t)$: there is an $a$ in $\sigma$ within interval $l$ from $t$

$\sigma$ $b$ $a$ $a$ $=$ $\diamond_a^{[1,2]}$
Input-determined automata and logics

Input-determined automata use $\Delta^I$ as guards and invariants

Input-determined logics modalities using $\Delta^I$

$\text{TMSO}^c: \Delta^I(x) \mid Q_a(x) \mid x \in X \mid x < y \mid \neg \varphi \mid \varphi \lor \varphi \mid \exists x \varphi \mid \exists X \varphi$

$\text{TLTL}^c: \Delta^I \mid a \mid \varphi U \varphi \mid \varphi S \varphi \mid \neg \varphi \mid \varphi \lor \varphi$
Outline

1. Verification and control of o-minimal systems
   - Motivations
   - Control of o-minimal systems
   - Weighted o-minimal hybrid systems

2. Expressivity and control for MTL
   - Real time logics
   - Expressivity of MTL and TPTL
   - Control for MTL specifications

3. Automata representation for timed logics
   - Existing results
   - Input-determined automata and logics
   - Expressiveness results
Timed logics and automata framework

Picture for timed logics (FSTTCS’06)

Automata and logical framework for the continuous semantics.

rec-TMSO$^c$

rec-CIDA
Timed logics and automata framework

Picture for timed logics (FSTTCS’06)

Automata and logical framework for the continuous semantics.
Timed logics and automata framework

Picture for timed logics (FSTTCS’06)

Automata and logical framework for the continuous semantics.

\( \text{rec-TMSO}^c \)
\( \text{rec-TFO}^c \)
\( \text{rec-TLTL}^c \)
\( \text{rec-CIDA} \)
Timed logics and automata framework

Picture for timed logics (FSTTCS’06)

Automata and logical framework for the continuous semantics.
Instanciating the operators

We can instanciate operators $\Delta = F, F^{-1}$

We obtain results on real-time logics
Instanciating the operators

We can instanciate operators $\Delta = F, F^{-1}$

We obtain results on real-time logics
Instanciating the operators

We can instanciate operators $\Delta = F, F^{-1}$
We obtain results on real-time logics

- rec-TMSO$^c$
- rec-TFO$^c$
- rec-TLTL$^c$
- rec-TLTL
- MTL+Past
- rec-CIDA
- rec-CFCIDA
- rec-TLTL
- rec-TLTL$^c$
- rec-TFO$^c$
- rec-TMSO$^c$
Instanciating the operators

We can instanciate operators \( \Delta = F, F^{-1} \)

We obtain results on real-time logics and applications!

Application: ultimate stability of MTL+Past
Conclusion
Conclusion

- O-minimal hybrid systems
  - control
  - optimal control
  - model-checking weighted systems

- Timed logics
  - decidability
  - control
  - expressivity
  - automata representation
Future work

- O-minimal hybrid systems
  - other winning conditions (TATL)
  - control under partial observability

- Timed logics
  - control for more tractable logics (MITL, Bounded-MTL, ···)
  - TPTL: do we need more than one clock?
  - other applications of continuous input-determined automata
Thank you for your attention!