Deciding Continuous-time Metric Temporal Logic with Counting Modalities

RP 2013

Marcello M. Bersani
Matteo Rossi
Pierluigi San Pietro
-
Politecnico di Milano

Motivations

- **Continuous time** is often used for modeling hybrid systems
  - computer systems that interact with the physical world
- Also well suited to capture asynchrony in systems
  - e.g., events occurring close to each other, but not at the same time
- Successful formalisms and tools to capture and analyze continuous systems
  - e.g., Timed Automata (Uppaal)
- Continuous-time **temporal logics** are useful to capture the properties of systems
  - e.g., high-level requirements
  - **descriptive** models of systems: what vs how

Motivations

- Since ’80s attempts to embed explicit (real) time in Linear Temporal Logic (LTL, defined on discrete time only)
  - Harel, Pnueli, Ostroff, etc.
- “A Really Temporal Logic” (**TPTL**), Alur&Henzinger, 1989, with explicit clocks
  - But undecidable over dense time
- Shortly after, Metric Temporal Logic (**MTL**) A&H ‘90.
  - Also undecidability over dense time
  - No explicit clocks, but implicit use of time in parameterized modalities $\Diamond_{<c}$
  - Decidable fragment: Metric Interval Temporal Logic (MITL) A&H ‘96

Example: MTL (and MITL)

$\phi_{MTL} = p \mid \neg\phi \mid \phi \wedge \phi \mid \phi U_I \phi \mid \phi S_I \phi$

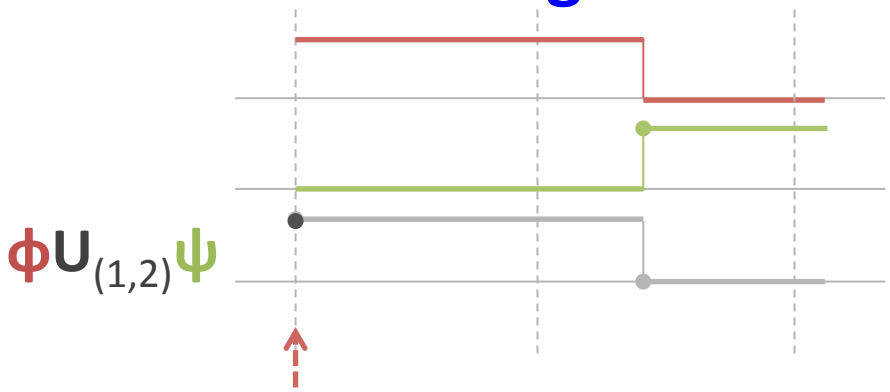
$I = \langle a,b \rangle$ or $\langle a,\infty \rangle$ $\quad a<=b \in \mathbb{N} \quad (\text{or } \mathbb{Q}_{\geq 0})$

p atomic proposition in finite alphabet AP

MITL = fragment of MTL with a<b (non punctual intervals)

M: $\mathbb{R} \longrightarrow 2^{AP}$

- Semantics over non-Zeno **Signals**



$\exists d' \in (1,2)$, $M,t+d' \models \psi$ and $M,t'' \models \phi$, $\forall t'' \in (t,t+d')$

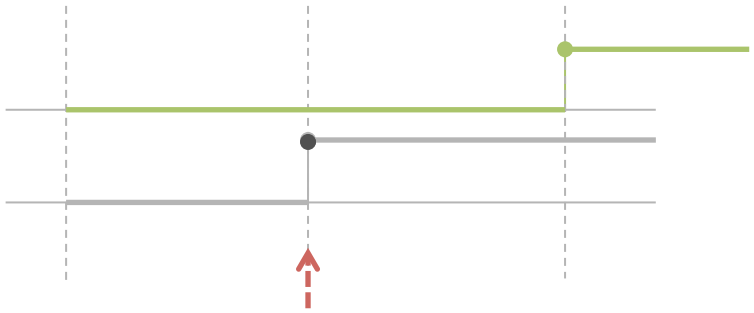
QTL (Quantitative Temporal Log.)

$$\phi_{QTL} = p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \mathbf{U} \phi \mid \phi \mathbf{S} \phi \mid \mathbf{F}_{(0,1)}\phi \mid \mathbf{P}_{(0,1)}\phi$$

- Semantics over non-Zeno (finitely variable) Signals

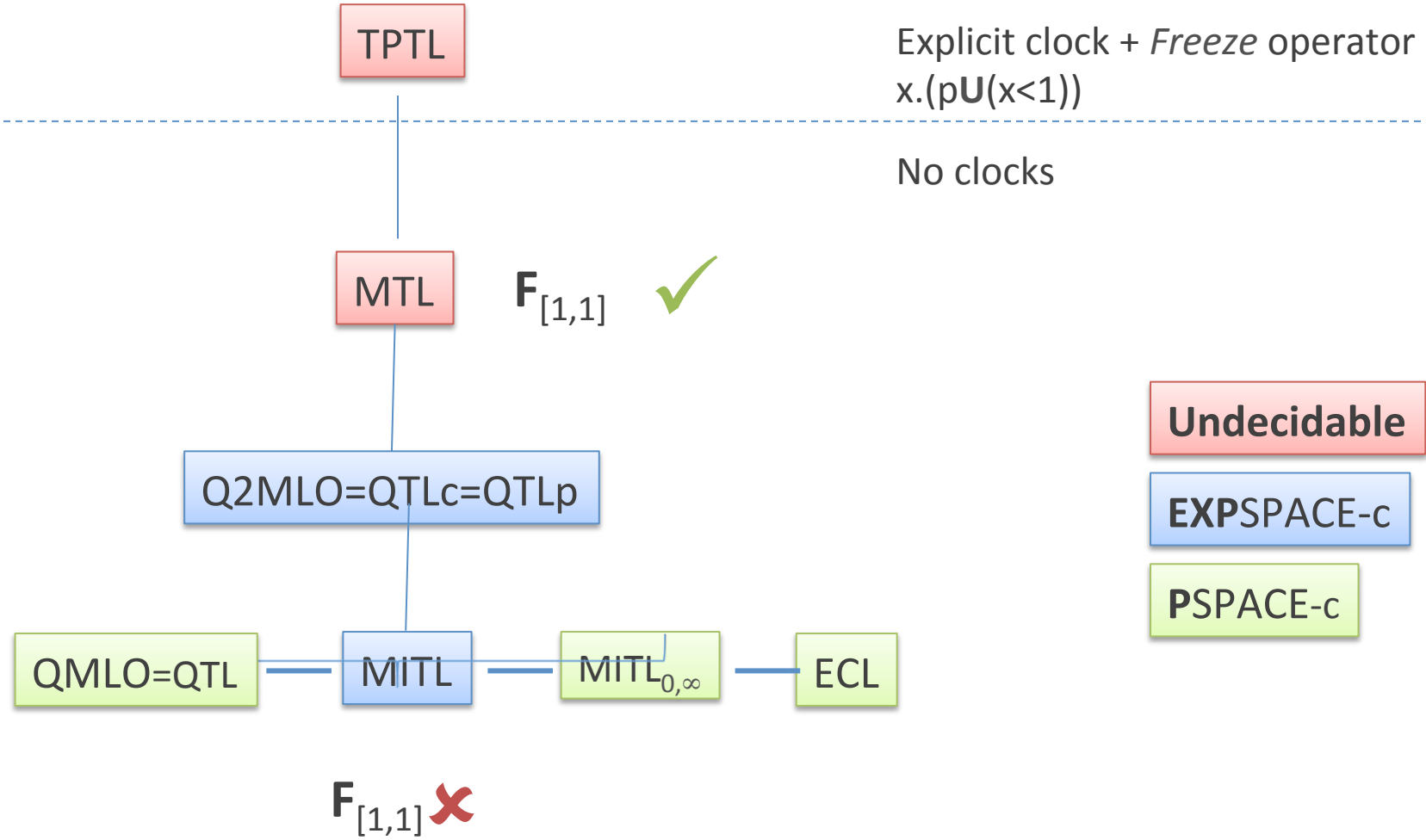
$M: \mathbb{R} \longrightarrow 2^{AP}$

$\mathbf{F}_{(0,1)}\phi$



- QTL has the same expressive power of MITL (Hirshfeld &Rabinovich '99)

Overview of SAT and inclusion for various continuous time logics



Pnueli modalities

- Pnueli conjectured that QTL and MITL are unable to express:
  «A and B will both happen within 1 time unit»
- Later proved by H&R '07, who generalized to any number n of events, required to occurr in order:
- **Pnueli modality**

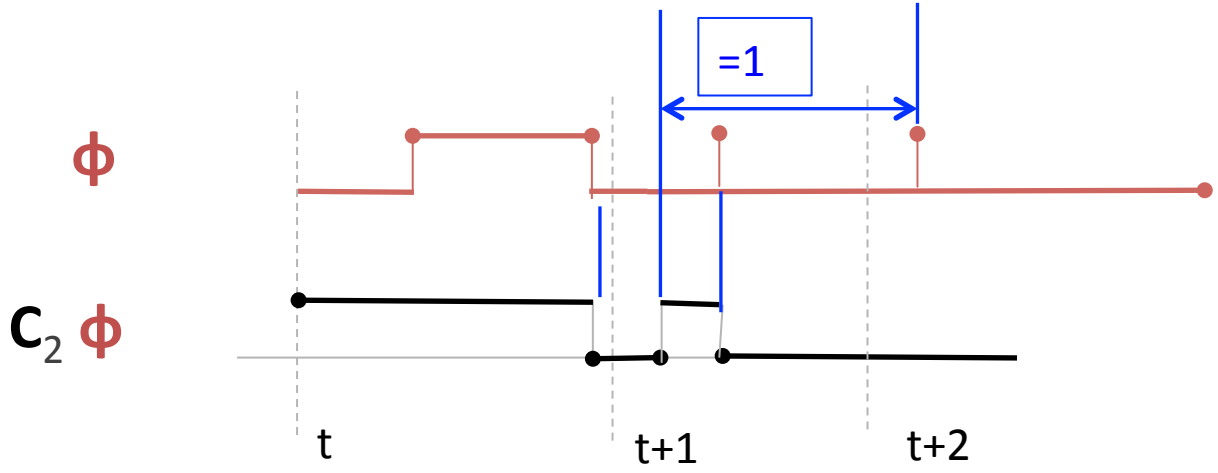
$\text{PNUELI}_n(\theta_1, \theta_2, .., \theta_n)$ true at instant t

iff there exist $t<t_1<\ldots<t_n<t+1$ s.t. each θ_i holds at t_i

Counting modalities

- H&R ‘12 defined a simpler **counting modality**:

$C_n(\phi)$ holds at instant t iff
ϕ holds at least n times in interval (t,t+1)



Background on Pnueli and Counting Modalities

- QTLc = QTL with (infinite) counting modalities
- QTLp = QTL with (infinite) Pnueli modalities

$$\text{QTLc} \equiv \text{QTLp}$$

- There was no tool supporting QTL or QTLc (until now!)

Existing tool for QTL

- Recently, we developed and implemented a **tool** deciding SAT for QTL
  - and all the equivalent real-time logics MITL, ECL, QMLO.
  - On unrestricted (non Zeno) signals
  - By using a new decidability procedure

- Now extended to deal with QTLc

[1] "From MITL to Timed Automata", Maler, Nickovic and Pnueli, 2006

Sketch of our solution

QLTc⟶ CLTL-over-clocks[1]

- CLTL-oc is a **discrete time** logic
- CLTL-oc decidable (**PSPACE-c**)
- CLTL-oc formulae contain **explicit clocks**
- Decision procedure
  - Based on (PSPACE) SAT of CLTL[2]
  - Using SMT tools (for solving bounded SAT)[3]

[1] "A Tool for Deciding Continuos Time Metric Temporal Logic", Bersani, Rossi, San Pietro, 2013
[2] "An automata Theoretic Approach to Constraint LTL", Demri, D'Souza, 2003
[3] "Constraint LTL Satisfiability Checking without Automata", Bersani et al., 2012

CLTL: Constraint LTL

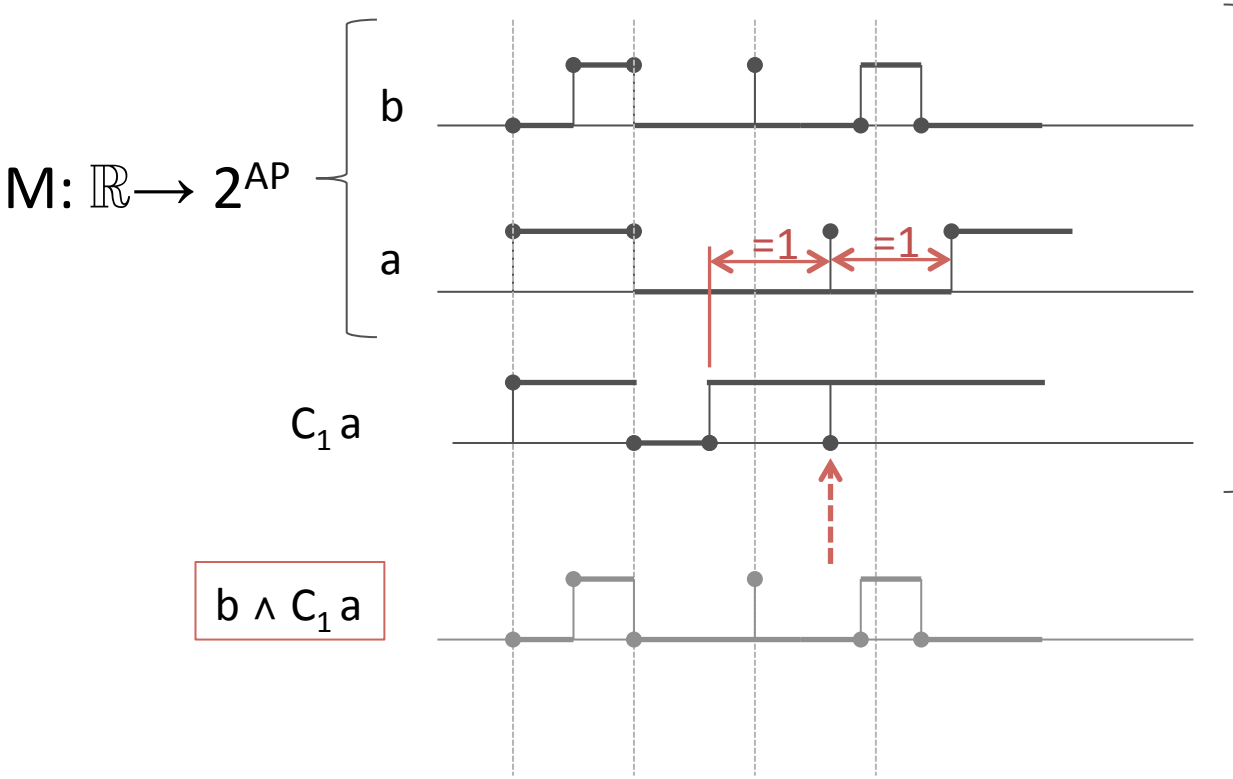
- Constraint LTL [Demri et al., 2006] is an extension of LTL where atomic propositions may be replaced by assertions (constraints) on the value of variables, e.g., x>0, x<y.
  - The type of variables and the kinds of allowed constraints lead to different logics.
  - The idea is interpreting variables over a **constraint system**.
  - A constraint system is pair <Domain, Relations>, e.g., $(\mathbb{N},<,=)$.
- Depending on the constraint system, the resulting logic may still be decidable.

Constraint LTL over clocks

- CLTL-oc is extension of CLTL with real variables behaving as Alur& Dill (timed automata) clocks.
- **AP=**finite set of propositions
  V= finite set of clocks (real variables)
- Syntax as in LTL: $\phi = \alpha \mid \neg\phi \mid \phi \wedge \phi \mid \phi \mathbf{U} \phi \mid \phi \mathbf{S} \phi \mid \mathbf{X}\phi \mid \mathbf{Y}\phi$
- But an atomic formula $\alpha = \mathbf{p} \mid \tau < \tau \mid \tau = \tau$ $\quad \mathbf{p} \in AP$
- Term τ: $\tau = \mathbf{c} \mid \mathbf{z} \mid \mathbf{X}\mathbf{z}$ constant $c \in \mathbb{N}$, clock $z \in \mathbb{R}$:
- Models: (π,σ) $\begin{cases} \pi: \mathbb{N} \rightarrow 2^{AP} \\ \sigma: \mathbb{N} \times \mathbf{V} \rightarrow \mathbb{R} \end{cases}$

From signals to CLTLoc models

- Given a QTLc formula, for every subformula θ let M_θ be the signal representing the changing points of θ



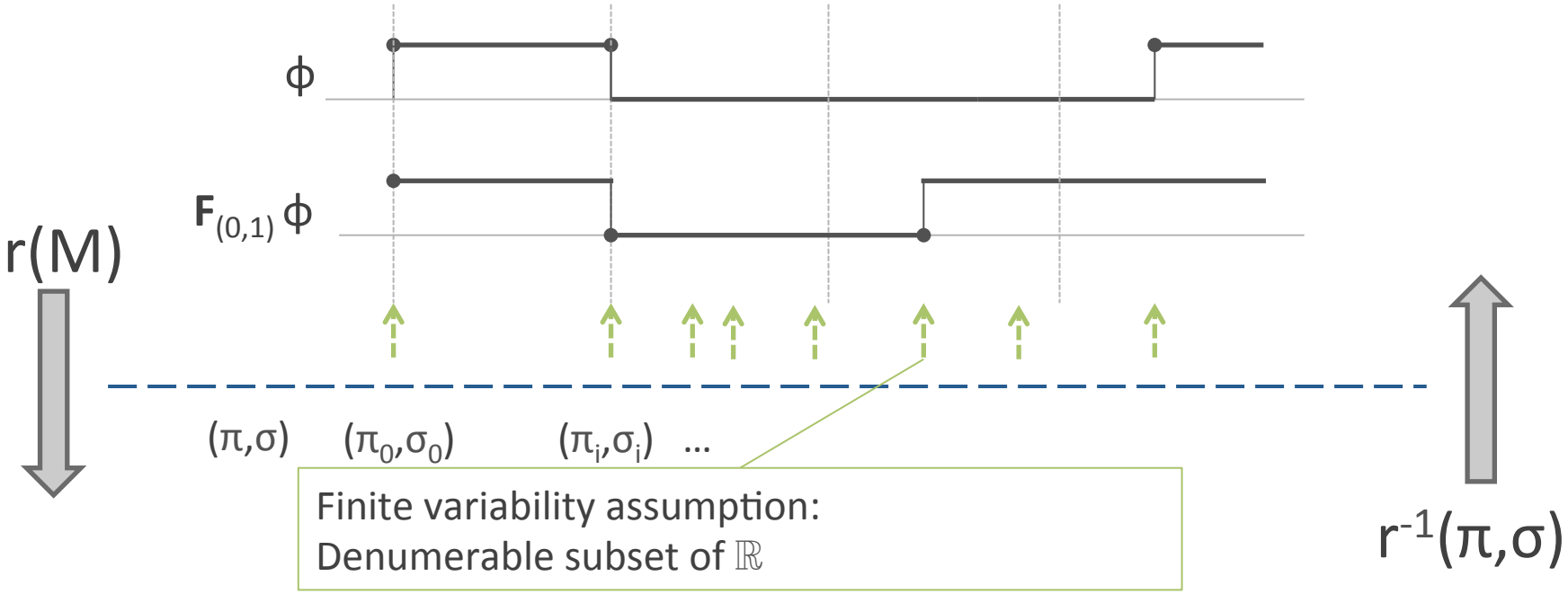
M_θ by QTL semantics

Finite variability assumption: Changing points are denumerable subset of ℝ

From signals to CLTLoc models

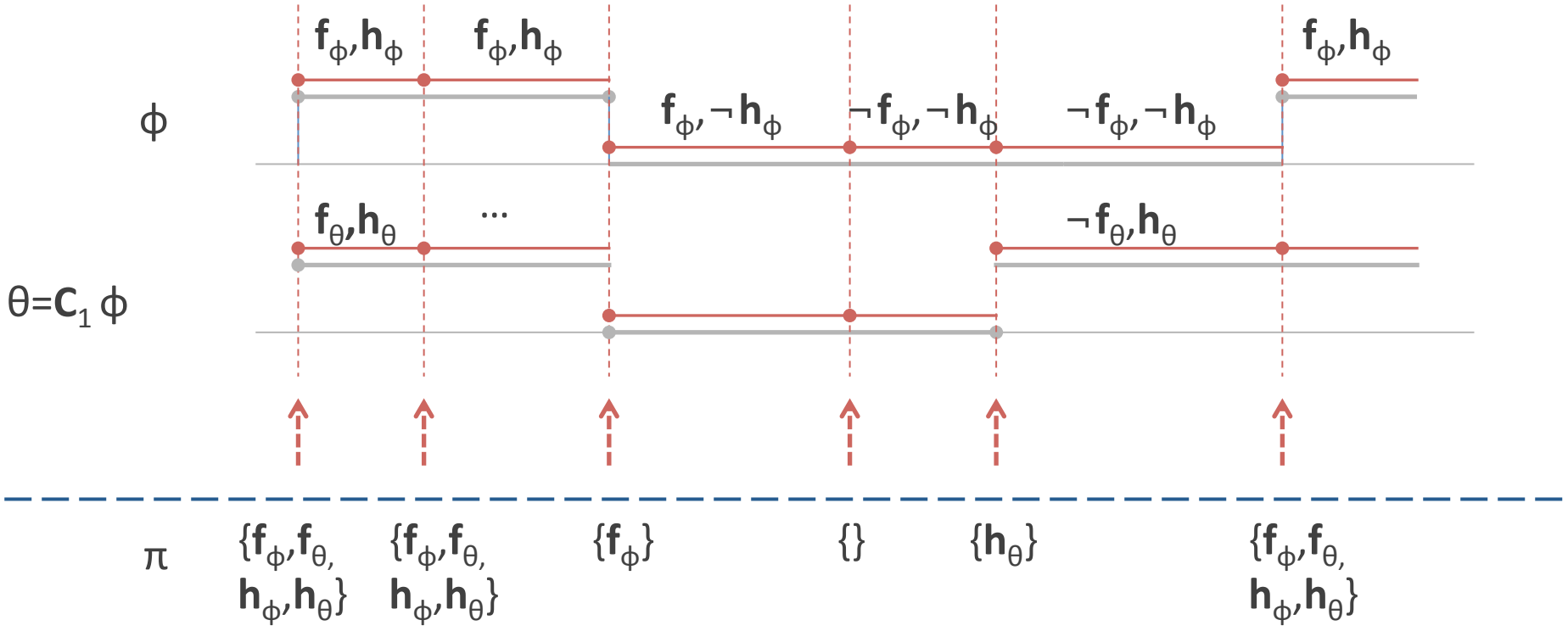
- Relation from signals M to CLTLoc models (π,σ)

  Changing point in M = a time instant in CLTLoc model



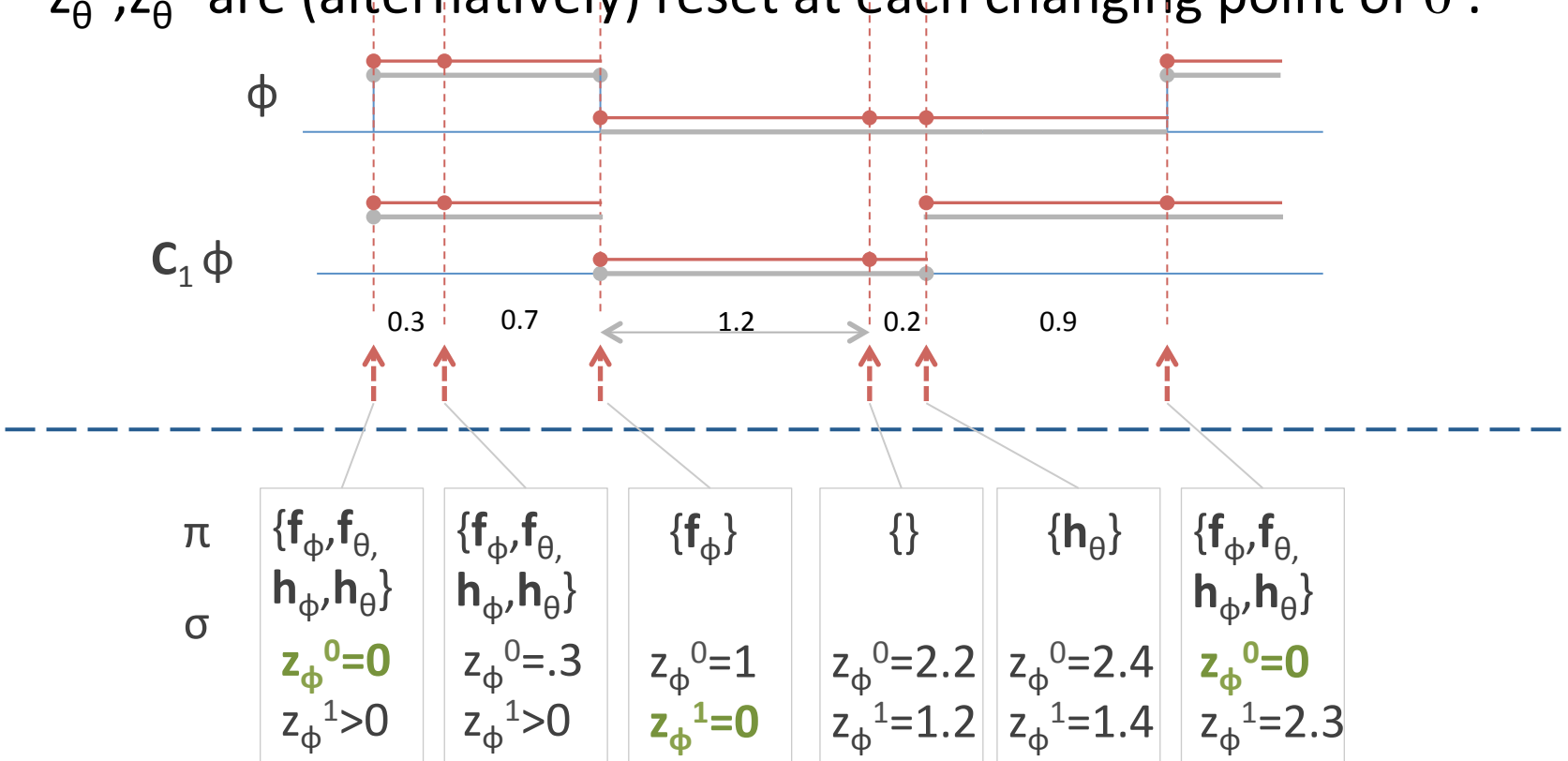
From signals to CLTLoc models

- Each position in π represents the truth of θ at the corresponding interval in M_θ
  - if atom $\mathbf{f}_\theta$ is true, θ *holds* in the first point of the current interval
  - if atom $\mathbf{h}_\theta$ is true, θ *holds* in the rest of the points of the current interval



From signals to CLTLoc models

- Instants in CLTLoc are always at distance one.
- Actual time progress between two instants is measured by **clocks**
- z_θ^0, z_θ^1 are (alternatively) reset at each changing point of θ .



Equisatisfiability

- Given a QTLc formula Φ, we define set of equisatisfiable CLTLoc formulae

$$\{m(\theta) \mid \theta \text{ subformula of } \Phi\}$$

such that

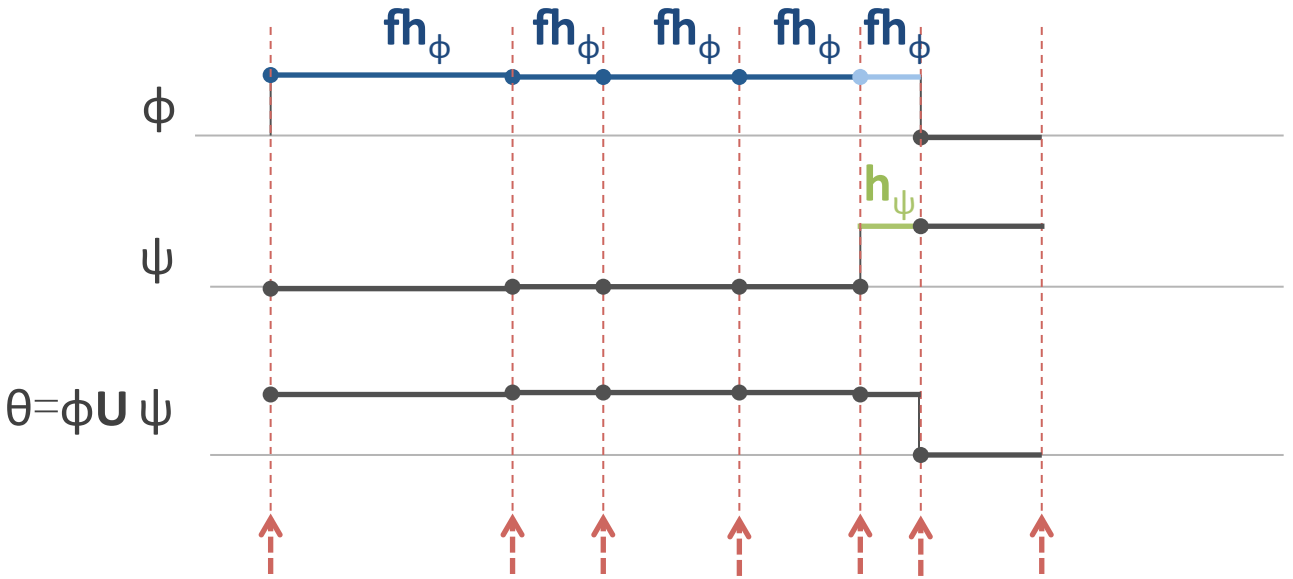
$$M,0 \vDash \Phi \quad \text{iff} \quad (\pi,\sigma),0 \vDash \mathbf{f}_{\Phi} \wedge \bigwedge_{\theta} \mathbf{G}(m(\theta))$$

(for all $(\pi,\sigma) \in r(M)$)

Example Translation for **U**

Left closed interval:

$fh_\phi = f_\phi \wedge h_\phi$

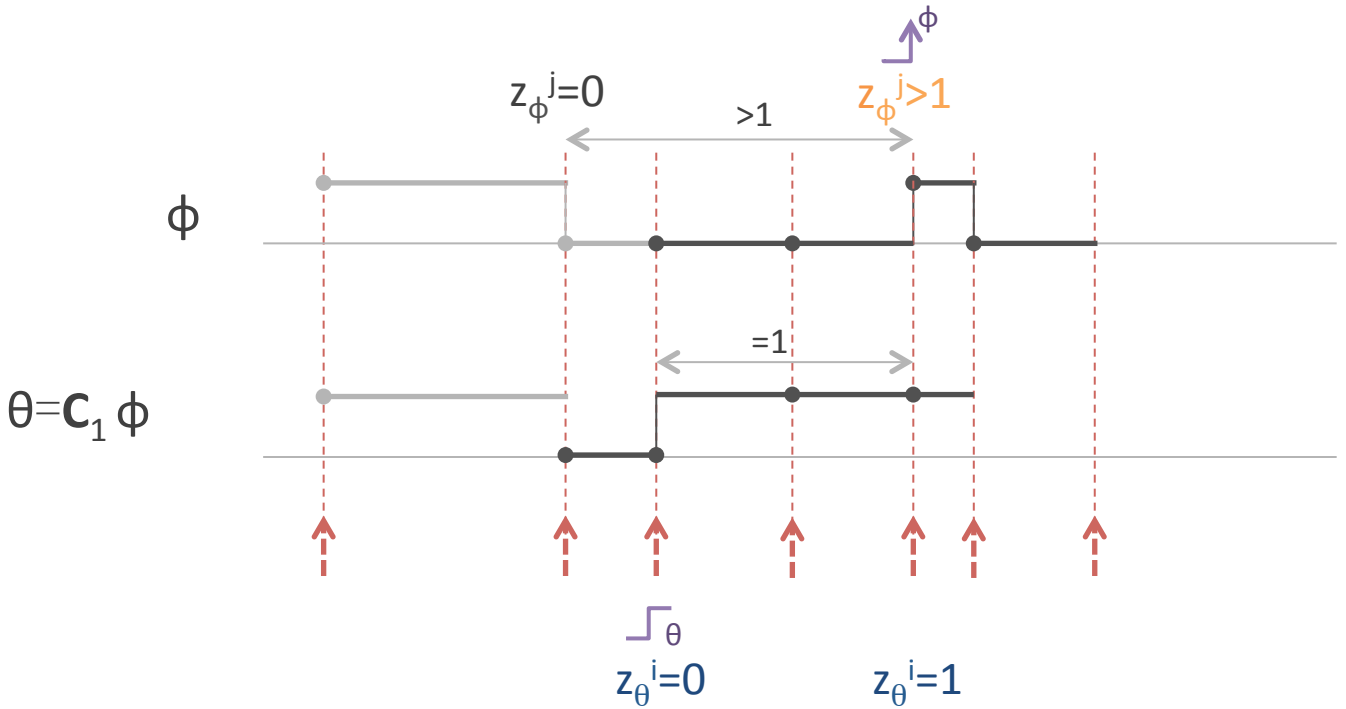


$m(\theta)$: $f_\theta \Leftrightarrow h_\theta$

$\wedge$

$h_\theta \Leftrightarrow h_\phi \wedge (h_\psi \vee X(fh_\phi U ((fh_\phi \wedge h_\psi) \vee f_\psi)))$

Translation for $\mathbf{C}_1$ (with 2 clocks)



$$m(\theta): \ \sqcap_\theta \Leftrightarrow \neg \mathbf{f}_\theta \wedge z_\theta^i=0 \wedge \mathbf{X}(z_\theta^i>0 \ \mathbf{U} \ (\uparrow\!\phi \wedge z_\theta^i=1 \wedge z_\phi^j>1))$$

$$\sqcap_\theta \Leftrightarrow \neg \mathbf{Y}(h_\theta) \wedge h_\theta$$

Generalization to $\mathbf{C}_n\,\phi$

- More Clocks: **n pair of clocks for each subformula of ϕ**
  - If using only QTL operators, one pair of clocks for every subformula is enough
    - Pairs of clocks are «recicled» between changing points
  - For counting modality $\mathbf{C}_n\,\phi$, it is necessary to keep track of up to n changes in the interval for ϕ

- Definition in CLTLoc of the various cases in terms of clocks and f,h
  - Classification of all possible cases for ϕ being true at least n times
    - Update rules for clocks
    - Raising (and falling) signals
    - Closed/open intervals
    - Singularities (isolated points)
    - In the origin of time axis
    - …

- Translation is «complicated», but it is conceptually easy within the CLTLoc framework

Example of a case for $\theta = \mathbf{C}_n\, \gamma$

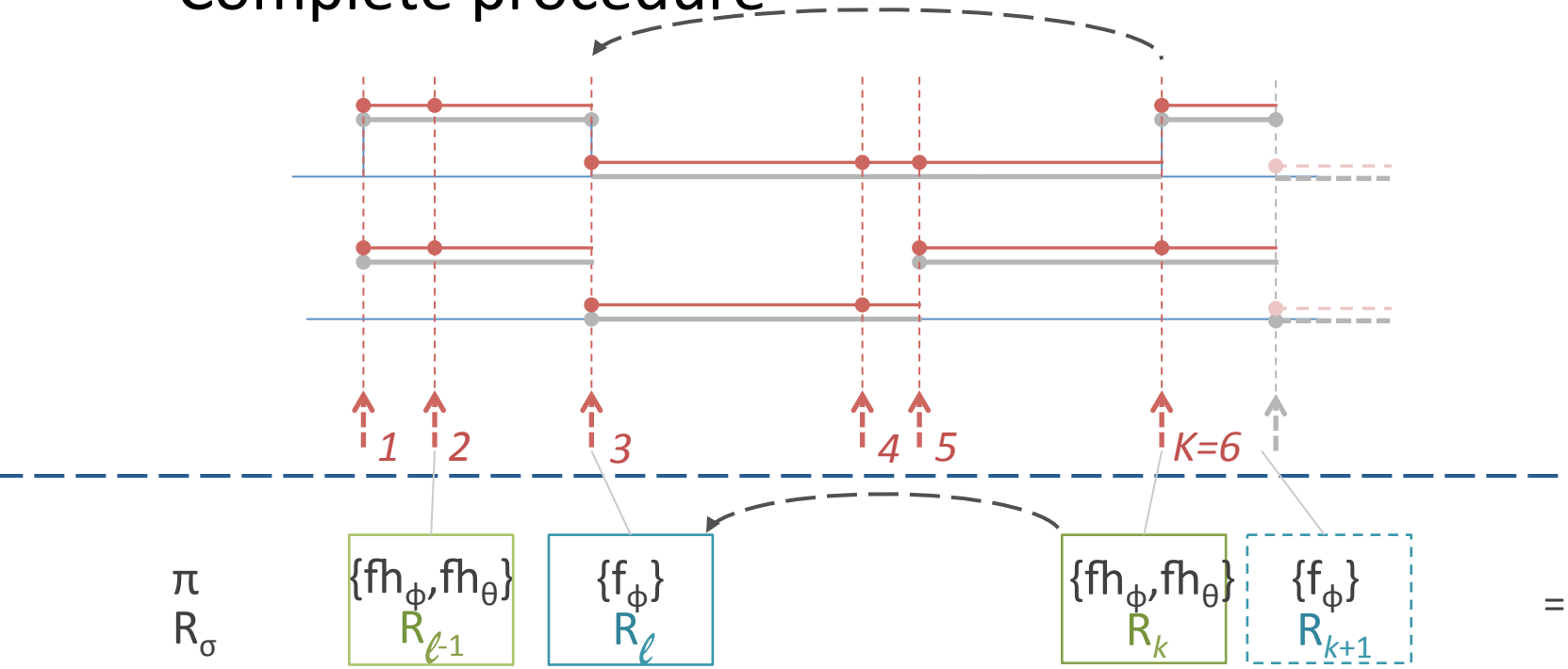
- "when θ becomes true with a raising in an instant t>0 then it does so in a left-open manner, a clock z^j_θ is reset, and
  - (i) either γ has n-1 up-singularities before z^j_θ hits 1 and γ becomes true again also with an up-singularity when $z^j_\theta = 1$, or
  - (ii) γ has a raising edge when $z^j_\theta = 1$ and it also has up to n-1 (possibly 0) up-singularities before $z^j_\theta = 1$"

Satisfiability checking of CLTL-oc

- CLTL-oc can be encoded into a decidable Satisfiability Modulo Theory (SMT) problem [Gandalf2013, AVOCS 2013]
  - Based on building a finite **symbolic** representaton of an ultimately **symbolic** model for a formula
  - The satisfiability is decided by solving at most a bounded amount of satisfiability problems of a decidable constraint system.
    - «reduction to the satisfiability problem of the theory of Equality and Uninterpreted Functions combined with Linear Integers/Reals Arithmetic" (QF-EUF ∪LIRA).
  - SMT solvers (e.g., Z3) can then be used to check satisfiability of CLTL-over-clocks.
  - The procedure is complete.

Verification: K-bounded SAT

- Find a (infinite) periodic model over
  - **Subformulae**
  - **Regions** for clocks (not over values!!)
- with *at most **K*** changing points
- Complete procedure



$= 12(3456)^{\omega}$

Complexity

- Satisfiability of QTLc is:
  - PSPACE-complete when indexes of counting modalities are encoded in unary
  - EXPSPACE-complete if indexes encoded in binary
- Translation from QTLc to CLTLoc
  - Polynomial in the size of the formula:
  - linear for formulae without counting modalites,
  - quadratic (in the unary encoding) for formulae inside counting modalities
- SAT of CLTLoc is PSPACE-complete[1]

Implementation

- The translation from QLTc (and also QTL, MITL) has been implemented in a new tool, qtlSolver
- Implementation is then based on
  - qtlSolver: http://code.google.com/p/qtlsolver/
    - Translation MITL (QTL) to CLTLoc
    - Java
  - ae^{2}Zot: arithmetical plugin for Zot
    - Bounded SAT for CLTL and CLTLoc
    - SMT based (Satisfiability Modulo Theory)

Simple Experiments

QTLc specification S

$$S = \mathbf{G}\left(\begin{array}{c} \mathbf{F}q \\ \wedge \\ q \rightarrow \mathbf{C}_2 q \end{array} \right)$$

MITL Properties P1 and P2

$$P1 = \mathbf{G}\left(\begin{array}{c} q \\ \rightarrow \\ \mathbf{F}_{(0,0.5)} q \end{array} \right)$$

$$P2 = \mathbf{G}\left(\begin{array}{c} q \\ \wedge \\ \mathbf{F}_{(0,0.5)} q \end{array} \right)$$

Formula	T	K
S	24s	25
S∧¬P1	50s	25
S∧¬P2	57m	25

- SAT
- UNSAT

Conclusions

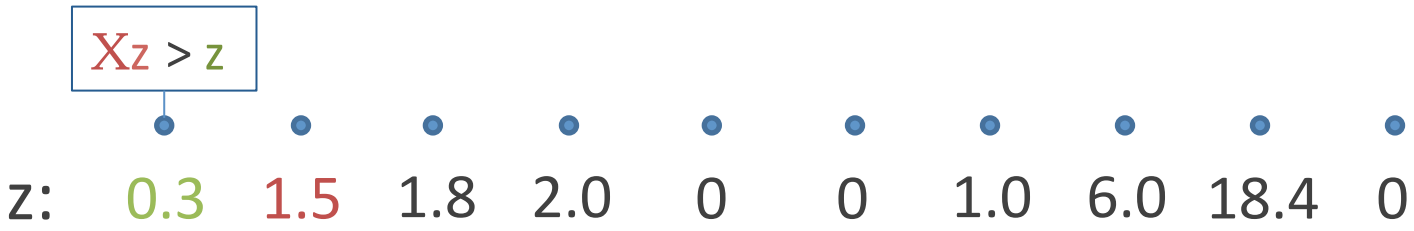
- CLTL-over-clocks can be considered as a target language to reduce decision problems of various continuous-time formalisms
  - MITL, QTL and QTLc (this paper)
  - QTLp= QTLc, but QTLp could be given a (more efficient) direct translation
  - but in principle also Timed Automata or Timed Petri Nets.

- To the best of our knowledge, our approach is the first allowing an effective implementation of a fully automated verification tool for continuous-time metric temporal logics

The end!

Clocks

- "Alur&Dill" clocks (e.g., timed automata)
  - Nonnegative
  - strongly monotonic (except for "resets")



  - Clock progressiveness[1] (non Zeno signals)

$$\mathbf{G}(z \geq 0) \wedge \mathbf{G}(Xz=0 \vee Xz>z) \wedge (\mathbf{GF}(z=0) \vee \mathbf{FG}(z>max_z))$$

$$\mathbf{G}(\phi) = \neg\mathbf{F}(\neg\phi) = \neg(\top\mathbf{U}\neg\phi)$$

[1] "A Theory of Timed Automata", Alur, Dill , 1994