Timed Pattern Matching

Doğan Ulus

joint with T. Ferrere, E. Asarin, O. Maler and D. Nickovic

Verimag, University of Grenoble-Alpes May 6, 2015

Real-time systems

Real-time systems → Systems with **timing constraints** → Control / Communication / Biological / . . .

They are complex

- Extremely large (or infinite) state-spaces
- Functional equivalence between abstractions is an exception.

Verification of real-time systems

- **Simulation**-based techniques to reason about correctness/performance
- Only **some segments** of simulation behaviors are interesting.

Pattern Matching

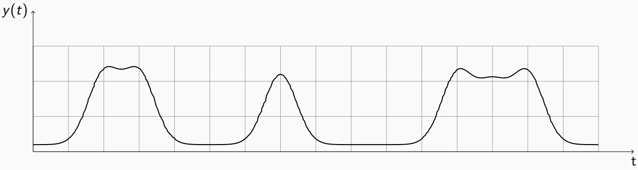
Example Find for "was" or "were" in the text

Regex Pattern $w(as + ere)$

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to Heaven, we were all going direct the other way - in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.

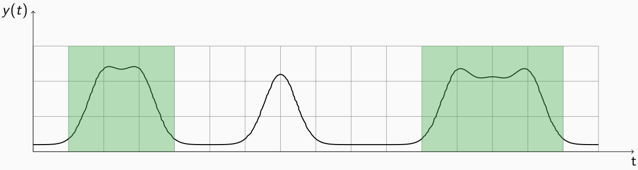
Our intention

Consider a simulation behavior including some pulses. Assume long pulses are interesting.



Our intention

Consider a simulation behavior including some pulses. Assume long pulses are interesting.

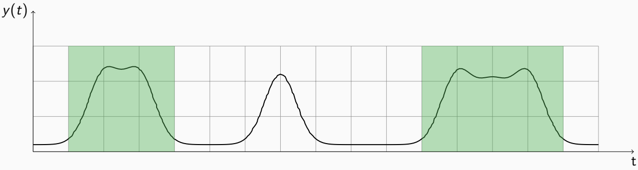


We would like to

+ **Locate** all interesting segments in a formal way.

Our intention

Consider a simulation behavior including some pulses. Assume long pulses are interesting.



We would like to

- **Locate** all interesting segments in a formal way.

How?

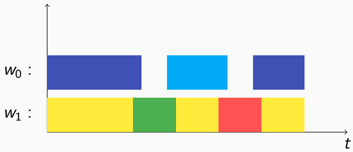
- Abstract behaviors in timed level
- Specify patterns using timed regular expressions
- Perform timed pattern matching

Outline

- A Long Introduction
  - Timed level of abstraction
  - Why not real-time logics?
  - Path to timed regular expressions
- Theory and Practice
  - Definitions
  - Algorithms
  - Implementation

Timed Level of Abstraction

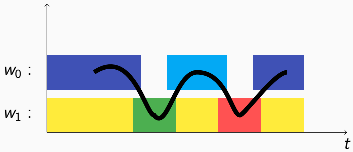
- Discrete values + Metric Time
- States as primitive timed entities



Visual representation for timed state sequences (signals)

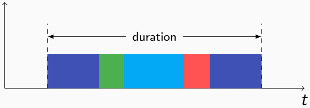
Timed Level of Abstraction

- Discrete values + Metric Time
- States as primitive timed entities



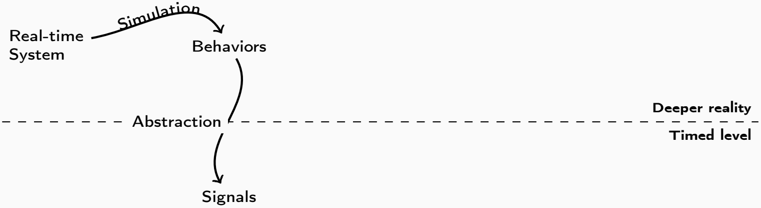
Visual representation for timed state sequences (signals)

- **Timed patterns** are meaningful compositions of timed states.
- Certain patterns are caused by design or by nature.

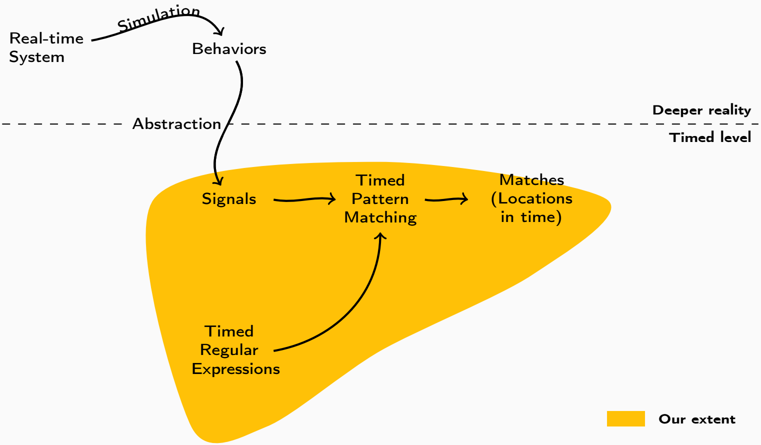


A timed pattern

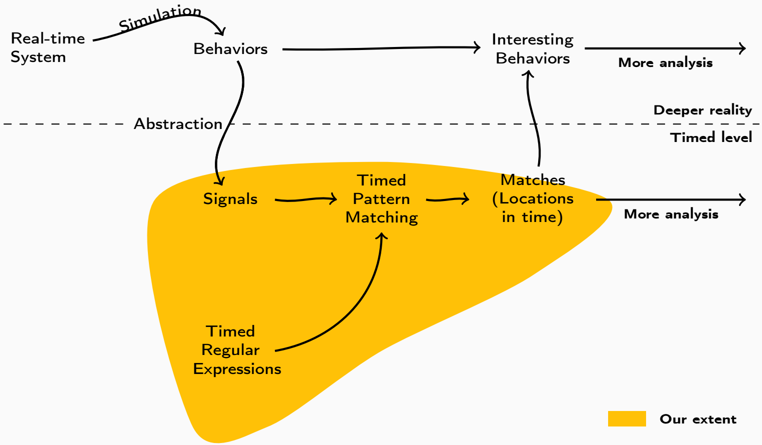
Flow


Real-time
System
Simulation
Behaviors
Abstraction
Deeper reality
Timed level
Signals

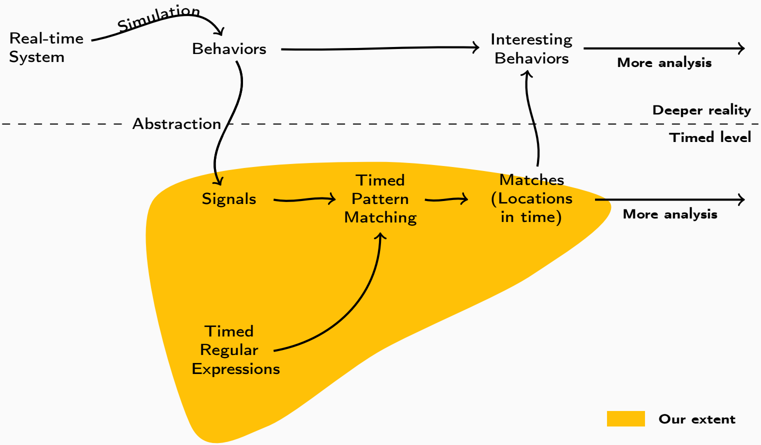
Flow


Real-time System
Simulation
Behaviors
Abstraction
Deeper reality
Timed level
Signals
Timed Pattern Matching
Matches (Locations in time)
Timed Regular Expressions
Our extent

Flow


Real-time System
Simulation
Behaviors
Interesting Behaviors
More analysis
Deeper reality
Abstraction
Timed level
Signals
Timed Pattern Matching
Matches (Locations in time)
More analysis
Timed Regular Expressions
Our extent

Flow

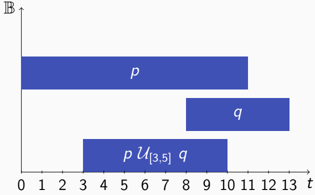


+ We use TRE as a timed specification language. Why not real-time logics?

Real-time logics

- Real-time logics (e.g. MTL) used to specify timed properties
- **Until** operator (of LTL) enhanced as $\mathcal{U}_I$ for time-bounded sequential reasoning.

$$(w, t) \vDash \psi_1 \mathcal{U}_{[a,b]} \psi_2 \leftrightarrow \exists t' \in [t + a, t + b].\ (w, t') \vDash \psi_2 \text{ and } \forall t'' \in [t, t'].\ (w, t'') \vDash \psi_1$$

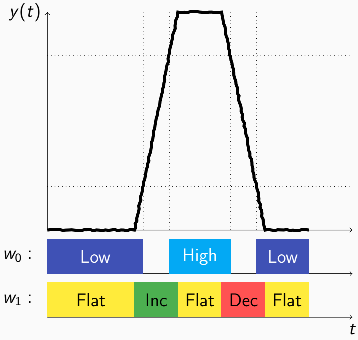


Pulse Example

+ Consider a pulse.
+ Pulse spec in English:

  When low, increase until high and flat more than 0.5 time units then decrease until low

$$w_0(t) = \begin{cases} \text{High} & \text{if } y(t) > c_h \\ \text{Low} & \text{if } y(t) < c_l \end{cases} \quad w_1(t) = \begin{cases} \text{Inc} & \text{if } \dot{y}(t) > d \\ \text{Dec} & \text{if } \dot{y}(t) < \text{-}d \\ \text{Flat} & \text{if otherwise} \end{cases}$$



+ In MTL:

$$\psi = (\text{Low} \wedge \text{Inc}) \; \mathcal{U} \; (\text{Inc} \; \mathcal{U} \; (\text{High} \wedge \text{Flat}) \; \mathcal{U}_{\geq 0.5} \; (\text{Dec} \; \mathcal{U} \; (\text{Dec} \wedge \text{Low}))))$$

+ In TRE:

$$\varphi := (\text{Low} \wedge \text{Inc}) \cdot \text{Inc} \cdot \langle \text{High} \wedge \text{Flat} \rangle_{\geq 0.5} \cdot \text{Dec} \cdot (\text{Dec} \wedge \text{Low})$$

Adding additional constraint over total duration will result:

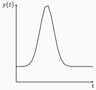
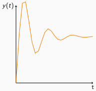
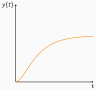
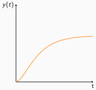
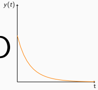
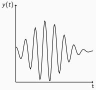
+ In MTL:

$$\psi' = (\mathsf{Low} \wedge \mathsf{Inc}) \ \mathcal{U} \ (\mathsf{Inc} \ \mathcal{U} \ (\mathsf{High} \wedge \mathsf{Flat}) \ \mathcal{U}_{\geq 0.5} \ (\mathsf{Dec} \ \mathcal{U} \ (\mathsf{Dec} \wedge \mathsf{Low})))) \wedge ((\mathsf{Low} \wedge \mathsf{Inc}) \vee \mathsf{Inc} \vee (\mathsf{High} \wedge \mathsf{Flat}) \vee \mathsf{Dec}) \ \mathcal{U}_{[2,5]} \ (\mathsf{Dec} \wedge \mathsf{Low})$$

+ In TRE:

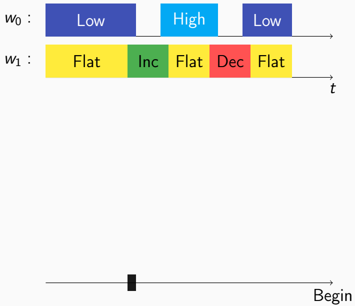
$$\varphi' := \langle (\mathsf{Low} \wedge \mathsf{Inc}) \cdot \mathsf{Inc} \cdot \langle \mathsf{High} \wedge \mathsf{Flat} \rangle_{\geq 0.5} \cdot \mathsf{Dec} \cdot (\mathsf{Dec} \wedge \mathsf{Low}) \rangle_{[2,5]}$$

Comparison 2 - Expressiveness

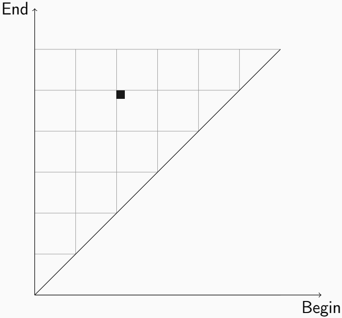
Everyday patterns		We can express in
	Sequential composition (Pulse)	MTL and TRE
 OR 	Alternation (2nd order response)	MTL and TRE
 AND 	Parallel composition (Switching capacitors)	MTL and TRE
	Repetition (Modulated pulse)	only TRE

Comparison 3 - Semantics

- MTL semantics is over **time-points**, monitoring gives only beginnings.
- TRE semantics is over **time-segments**, monitoring gives all beginnings, endings and durations.



$\chi(\psi, w)$



$\mathcal{M}(\varphi, w)$

Timed regular expressions

$$\varphi := \epsilon \mid p \mid \overline{p} \mid \varphi \cdot \varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi^* \mid \langle \varphi \rangle_I$$

p is a propositional variable, I is an interval

$$\begin{array}{lll}
(w, t, t') \vDash \epsilon & \leftrightarrow & t = t' \\
(w, t, t') \vDash p & \leftrightarrow & t < t' \text{ and } \forall t'' \in (t, t'),\ p[t''] = 1 \\
(w, t, t') \vDash \overline{p} & \leftrightarrow & \ldots \\
(w, t, t') \vDash \varphi \cdot \psi & \leftrightarrow & \exists t'' \in (t, t'),\ (w, t, t'') \vDash \varphi \text{ and } (w, t'', t') \vDash \psi \\
(w, t, t') \vDash \varphi \vee \psi & \leftrightarrow & (w, t, t') \vDash \varphi \text{ or } (w, t, t') \vDash \psi \\
(w, t, t') \vDash \varphi \wedge \psi & \leftrightarrow & \ldots \\
(w, t, t') \vDash \varphi^* & \leftrightarrow & \exists k \geq 0,\ (w, t, t') \vDash \varphi^k \\
(w, t, t') \vDash \langle \varphi \rangle_I & \leftrightarrow & t' - t \in I \text{ and } (w, t, t') \vDash \varphi
\end{array}$$

Formal problem statement

Definition (Match-set)

For a signal w and an expression φ the match-set is

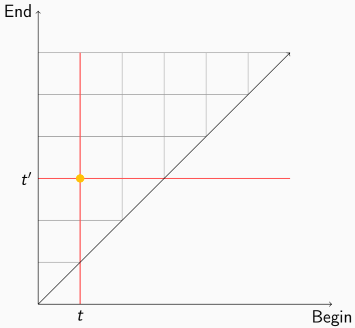
$$\mathcal{M}(\varphi, w) := \{(t, t') \mid (w, t, t') \vDash \varphi\}$$

Problem (Timed pattern matching)

Given a signal and an expression compute the match-set.

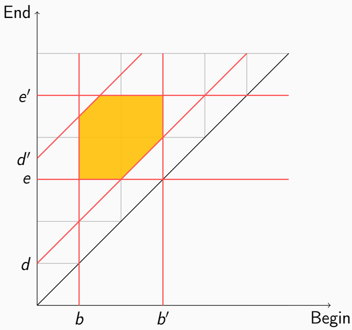
Data structure

Mark (t, t') if $(w, t, t') \vDash \varphi$.



A match beginning at t ending at t'.

Better mark as zones.



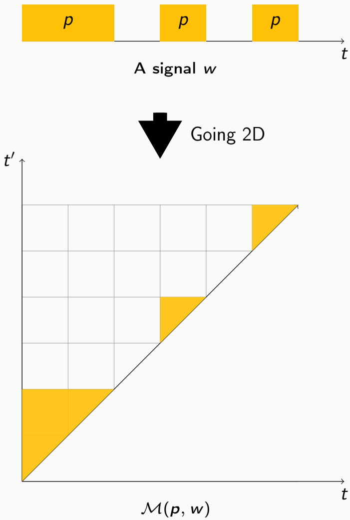
$$b \leq t \leq b'$$
$$e \leq t' \leq e'$$
$$d \leq t' - t \leq d'$$

Main result

Theorem

The match-set $\mathcal{M}(\varphi, w)$ is computable as a finite union of 2D zones.

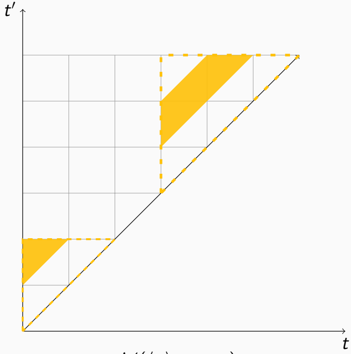
Base cases - Literals



- When a segment of p satisfies, all sub-segments satisfy p.
- Triangle zones

Base cases - Duration constraints

A signal w



$\mathcal{M}(\langle p \rangle_{[1,2]}, w)$

+ Restricting duration

$$\mathcal{M}(\langle \varphi \rangle_I, w) = \mathcal{M}(\varphi, w) \cap \{(t, t') \mid t' - t \in I\}$$

Base cases - Concatenation

- Concatenation is a composition of match sets.

$$\mathcal{M}(\varphi \cdot \psi) = \mathcal{M}(\varphi) \circ \mathcal{M}(\psi)$$

$$(t, t') \in \mathcal{M}(\varphi) \circ \mathcal{M}(\psi) \leftrightarrow \exists t'' : (t, t'') \in \mathcal{M}(\varphi) \wedge (t'', t') \in \mathcal{M}(\psi)$$

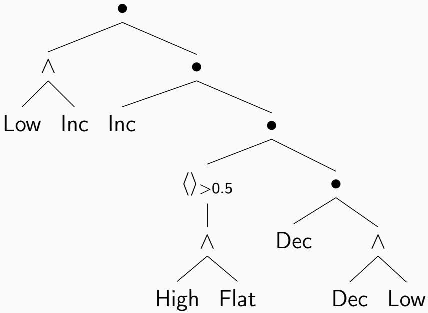
- Can be obtained using standard zone operations.
- Composition preserves zones and match sets

$$\bigcup_i z_i \circ \bigcup_j z'_j = \bigcup_{ij} z_i \circ z'_j$$

- Most resulting zones are empty in practice.
- *Plane-sweep* algorithm: sorting zones by start / end time allows to avoid most empty operations

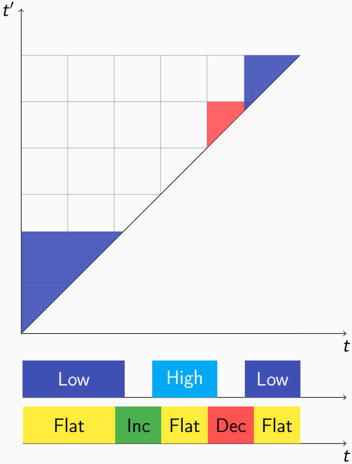
Overall Computation

$$\varphi := (\mathsf{Low} \wedge \mathsf{Inc}) \cdot \mathsf{Inc} \cdot \langle \mathsf{High} \wedge \mathsf{Flat} \rangle_{>0.5} \cdot \mathsf{Dec} \cdot (\mathsf{Dec} \wedge \mathsf{Low})$$



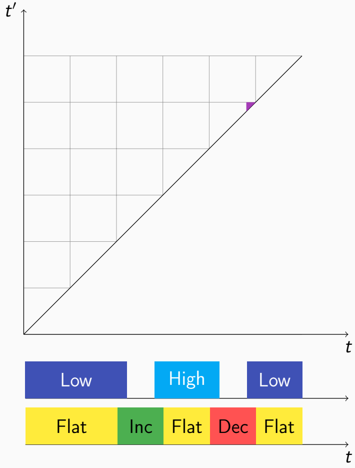
Matching Pulse Example

$$\varphi := (\text{Low} \wedge \text{Inc}) \cdot \text{Inc} \cdot \langle \text{High} \wedge \text{Flat} \rangle_{\geq 0.5} \cdot \text{Dec} \cdot (\text{Dec} \wedge \text{Low})$$



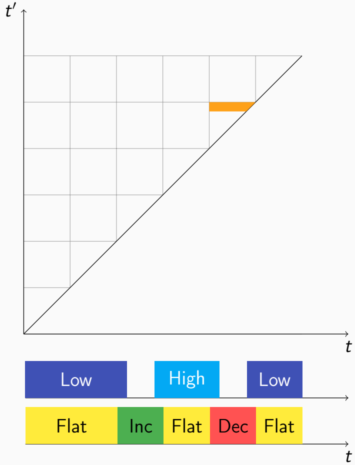
Matching Pulse Example

$$\varphi := (\mathsf{Low} \wedge \mathsf{Inc}) \cdot \mathsf{Inc} \cdot \langle \mathsf{High} \wedge \mathsf{Flat} \rangle_{\geq 0.5} \cdot \mathsf{Dec} \cdot (\mathsf{Dec} \wedge \mathsf{Low})$$



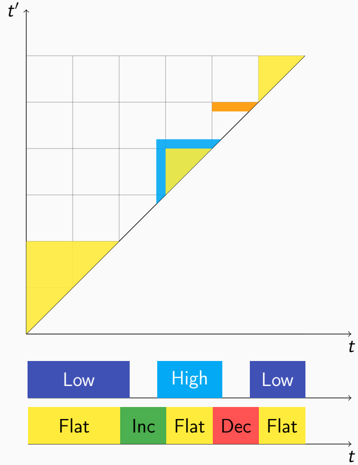
Matching Pulse Example

$$\varphi := (\mathsf{Low} \wedge \mathsf{Inc}) \cdot \mathsf{Inc} \cdot \langle \mathsf{High} \wedge \mathsf{Flat} \rangle_{\geq 0.5} \cdot \mathsf{Dec} \cdot (\mathsf{Dec} \wedge \mathsf{Low})$$



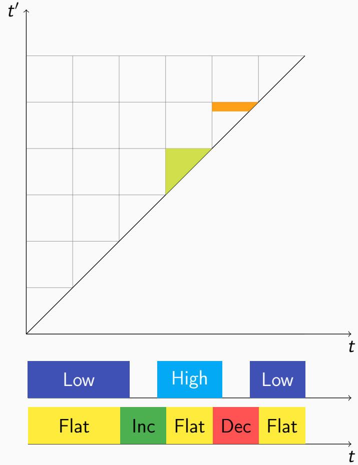
Matching Pulse Example

$$\varphi := (\mathsf{Low} \wedge \mathsf{Inc}) \cdot \mathsf{Inc} \cdot \langle \mathsf{High} \wedge \mathsf{Flat} \rangle_{\geq 0.5} \cdot \mathsf{Dec} \cdot (\mathsf{Dec} \wedge \mathsf{Low})$$



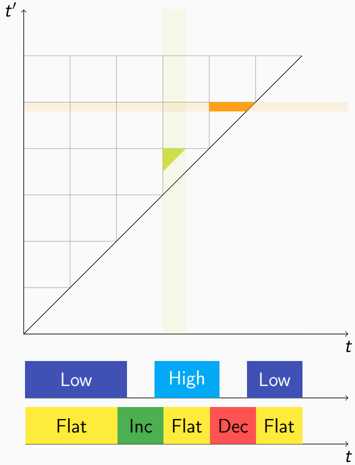
Matching Pulse Example

$$\varphi := (\text{Low} \wedge \text{Inc}) \cdot \text{Inc} \cdot \langle \text{High} \wedge \text{Flat} \rangle_{\geq 0.5} \cdot \text{Dec} \cdot (\text{Dec} \wedge \text{Low})$$



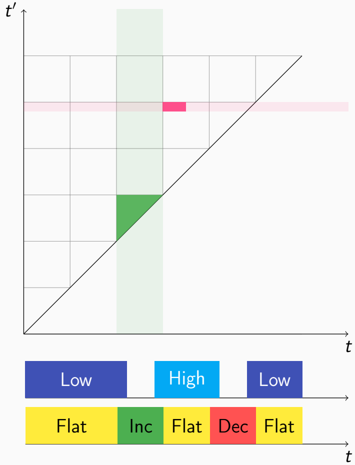
Matching Pulse Example

$\varphi := (\mathsf{Low} \wedge \mathsf{Inc}) \cdot \mathsf{Inc} \cdot \langle \mathsf{High} \wedge \mathsf{Flat} \rangle_{\geq 0.5} \cdot \mathsf{Dec} \cdot (\mathsf{Dec} \wedge \mathsf{Low})$



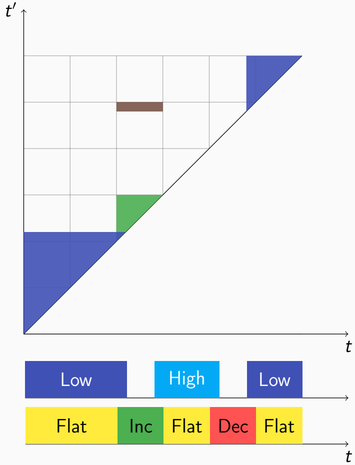
Matching Pulse Example

$$\varphi := (\text{Low} \wedge \text{Inc}) \cdot \text{Inc} \cdot \langle \text{High} \wedge \text{Flat} \rangle_{\geq 0.5} \cdot \text{Dec} \cdot (\text{Dec} \wedge \text{Low})$$



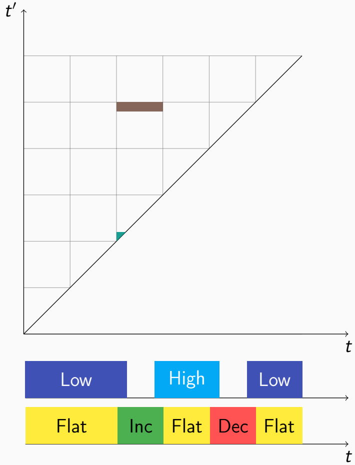
Matching Pulse Example

$$\varphi := (\text{Low} \wedge \text{Inc}) \cdot \text{Inc} \cdot \langle \text{High} \wedge \text{Flat} \rangle_{\geq 0.5} \cdot \text{Dec} \cdot (\text{Dec} \wedge \text{Low})$$



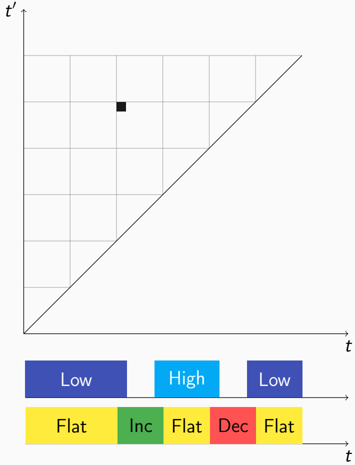
Matching Pulse Example

$$\varphi := (\mathsf{Low} \wedge \mathsf{Inc}) \cdot \mathsf{Inc} \cdot \langle \mathsf{High} \wedge \mathsf{Flat} \rangle_{\geq 0.5} \cdot \mathsf{Dec} \cdot (\mathsf{Dec} \wedge \mathsf{Low})$$



Matching Pulse Example

$$\mathcal{M}(\varphi, w) = \{\blacksquare\} = \{(t, t') \in [2, 2.2] \times [3.8, 4]\}$$



Return back to behaviors, segments in ■ contain a pulse.

Prototype Tool

On Implementation

- in Python and C (using IF library for zones)

On Performance

- 32K zones + complex expression = few seconds
- Negligible overhead compared to simulation times

Conclusion

+ TRE is intuitive, expressive and informative for timed pattern matching purposes.
+ Problem of timed pattern matching stated and solved in a 2D world.
+ A prototype tool developed.
+ Experiments on synthetic data witness scalability.

More details in

+ **Timed Pattern Matching**, [FORMATS'14]
  D. Ulus, T. Ferrere, E. Asarin, O. Maler
+ **Measuring with Timed Patterns**, [CAV'15]
  T. Ferrere, D. Nickovic, O. Maler, D. Ulus