PhD thesis defence: July 7th, 2023

Parametric Timed Formalisms for Specification and Monitoring

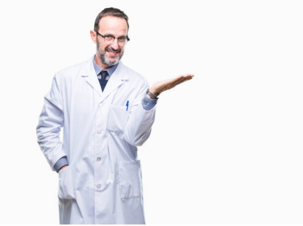
Akshay Mambakam

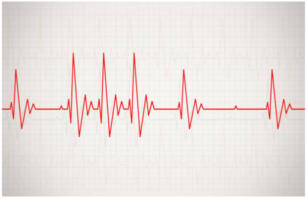
Supervisors: Nicolas Basset and Thao Dang (VERIMAG)

Jury:
Étienne André, Université Sorbonne Paris Nord, Rapporteur
Eugene Asarin, IRIF, Université Paris Cité, Invité
Ezio Bartocci, TU Wien, Rapporteur
Saddek Bensalem, VERIMAG, Université Grenoble Alpes, Examinateur
Patricia Bouyer-Decitre, LMF, CNRS, Examinatrice
Dejan Ničković, Austrian Institute of Technology (AIT), Examinateur
Nicola Paoletti, King's College London, Examinateur

Introduction: First Example

- Electrocardiogram (ECG): Electrocardiogram signals capture information about electrical activity of the heart and can help detect anomalies in its functioning
- Doctors read the signal and detect problems



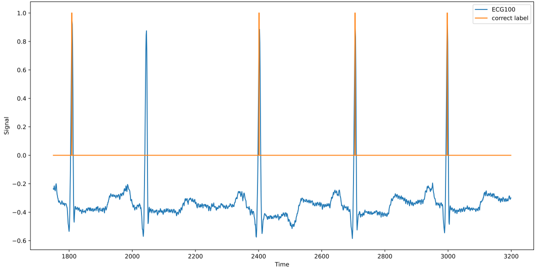


Help doctors by automating

Introduction: First Example

Possible solutions for automation

- Neural networks. Black-box. Not explainable
- Formal specifications. Interpretable and rigorous

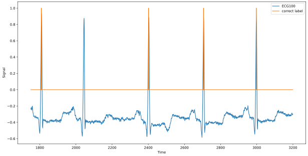


Something wrong in the ECG

Introduction: First Example

Possible solutions for automation

- Neural networks. Black-box. Not explainable
- Formal specifications. Interpretable and rigorous



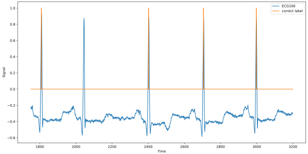
Explaining above ECG formally

- Pulse arrives prematurely (anomalous)
- Normal beat: `pulse` and (`eventually in`$_{[0,316]}$ `pulse`)
- `pulse` : `height` $\geq$ 1.24 on a time `window` of width 44 and `variation` less than 0.49 outside

Introduction: First Example

Possible solutions for automation

- Neural networks. Black-box. Not explainable
- Formal specifications. Interpretable and rigorous



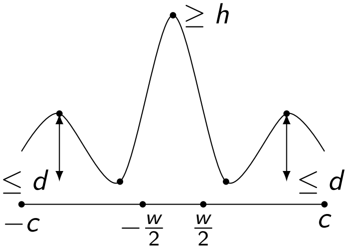
Explaining above ECG formally

- Pulse arrives prematurely (anomalous)
- Normal beat: `pulse` and (`eventually in`$_{[0,316]}$ `pulse`)
- `pulse` : `height` $\geq \cancel{1.24}$ (h) on a time `window` of width $\cancel{44}$ (w) and `variation` less than $\cancel{0.49}$ (d) outside

Introduction: Parameters

Parameter learning

Expert (doctor) gives shape, we learn parameters that fit real (ECG) data.



Depending on the patient and the scaling the parameters differ

Introduction: General Context

- Cyber-Physical Systems (CPS): Computer system + Physical environment
- Monitoring, runtime verification, trace analysis for CPS
  - Given a trace of a system detect special fragments (dangerous manoeuvres, arrhythmia, circadian rhythms, a melody)
  - On-line (train accident prevention) or off-line (plane black box)

In This Thesis

We consider two parametric timed formalisms for specification and monitoring:

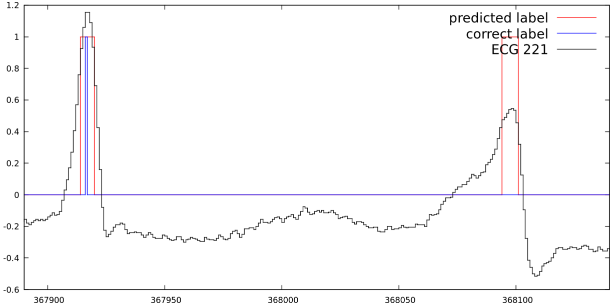
- Parametric Signal Temporal Logic (and extensions)
- Parametric Timed Regular Expressions

Outline

Learning Parameters for STL Specifications

Our Learning Framework

- Patterns in time series: e.g. ECGs
- *Labelling* of patterns: given by experts (doctors)
- Pattern predictor: STL formula which gives *prediction*
- Class of pattern predictors: parametric STL formula
- Learning: Find parameters such that prediction ≈ labelling



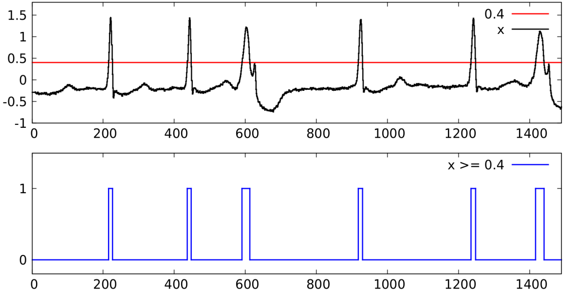
Outline

1. Learning Parameters for STL Specifications
   - Background: Signal Temporal Logic (STL)
   - STL for Accurate Pattern Prediction
   - Monotonic Formulae
   - Algorithm
   - Experiments

Background: Signal Temporal Logic

From [Maler and Ničković, 2004]. Deals with:

- Real-valued signals (x)
- Boolean signals ($x \geq 0.4$)



Background: Signal Temporal Logic

Predicates for signals

$(x \geq c) \mid (x \leq c) \mid p$

Examples

- Speed of a vehicle is below the prescribed limit ($V \leq 40$)
- Indicating sleeping status (`True` or `False`) over time: `sleep`

Background: Signal Temporal Logic

Temporal operators

$F_{[a,b]}\varphi \mid G_{[a,b]}\varphi \mid \varphi_1 \; U_{[a,b]} \; \varphi_2$
Eventually, Always and Until

Examples

- Student finishes exam within one hour ($F_{[0,1]}$ `finish`)
- Student is always awake for two hours during a lecture ($G_{[0,2]}$ `awake`)
- Student does not look at others' sheet until the exam is over (($\neg$`look`)$U_{[0,\infty)}$`over`)

Background: Extended Signal Temporal Logic

Need for min/max operators

- We have a peak in ECG signal ($x \geq 2$). How to arrive at 2?
- ECG signal stabilizes ($-0.5 \leq x \leq 0.2$). What if we do not know the limits?

From [Bakhirkin and Basset, 2019]. Introduce (local) maximum/minimum operators

$$\mathrm{Max}_{[a,b]}\, x \;\mid\; \mathrm{Min}_{[a,b]}\, x$$

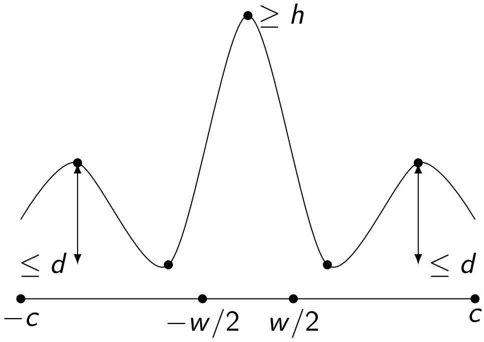
Outline

1. Learning Parameters for STL Specifications
   - Background: Signal Temporal Logic (STL)
   - STL for Accurate Pattern Prediction
   - Monotonic Formulae
   - Algorithm
   - Experiments

STL for Accurate Pattern Prediction

ECG pulse example:

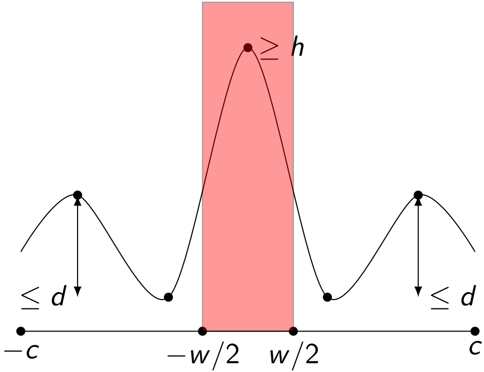
$$\Psi^{ch}_{(w,d,h)} := ((\mathrm{Max}_{[-c,-w/2]}\, s - \mathrm{Min}_{[-c,-w/2]}\, s) \leq d) \wedge$$
$$((\mathrm{Max}_{[-w/2,w/2]}\, s) \geq h) \wedge ((\mathrm{Max}_{[w/2,c]}\, s - \mathrm{Min}_{[w/2,c]}\, s) \leq d)$$



STL for Accurate Pattern Prediction

ECG pulse example: peak,

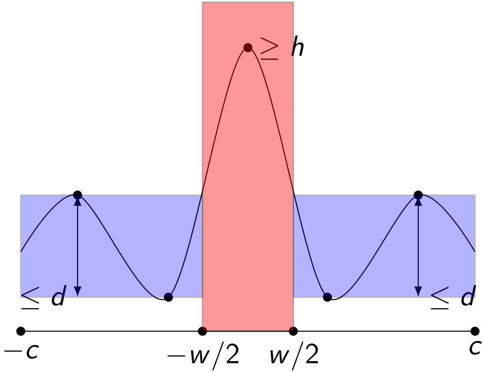
$$\Psi^{ch}_{(w,d,h)} := ((\mathrm{Max}_{[-c,-w/2]}\, s - \mathrm{Min}_{[-c,-w/2]}\, s) \leq d) \wedge$$
$$((\mathrm{Max}_{[-w/2,w/2]}\, s) \geq h) \wedge ((\mathrm{Max}_{[w/2,c]}\, s - \mathrm{Min}_{[w/2,c]}\, s) \leq d)$$



STL for Accurate Pattern Prediction

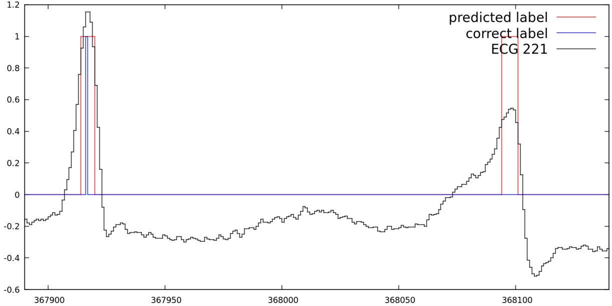
ECG pulse example: peak, stabilization

$$\Psi^{ch}_{(w,d,h)} := ((\mathrm{Max}_{[-c,-w/2]}\, s - \mathrm{Min}_{[-c,-w/2]}\, s) \leq d) \wedge$$
$$((\mathrm{Max}_{[-w/2,w/2]}\, s) \geq h) \wedge ((\mathrm{Max}_{[w/2,c]}\, s - \mathrm{Min}_{[w/2,c]}\, s) \leq d)$$



Learn to Predict Patterns

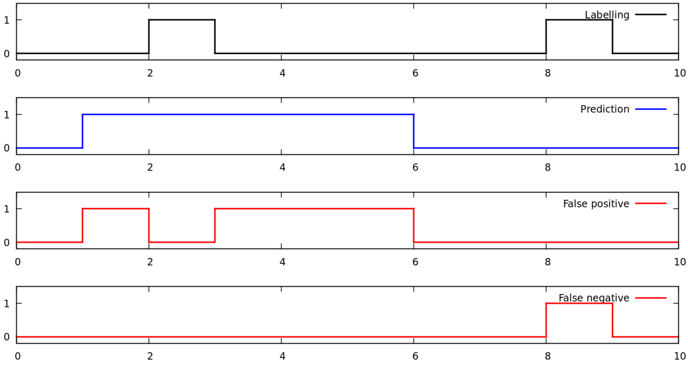
- Labelling and prediction: Both Boolean signals
- Pattern predictor (STL formula): Given observation (real-valued signals) gives predictions (Boolean signals)
- Prediction should match labelling



Defining Mismatch/Error: False Positives and False Negatives

- Class of pattern predictors from which to select: Given by a parametric STL formula
- Learn parameters while bounding mismatches w.r.t labelling
- "How often" does a mismatch occur? How to quantify?
- The <u>false positive signal</u> indicates when the predictor predicts an occurrence when there is none
- The <u>false negative signal</u> indicates when the predictor misses an actual occurrence

Illustration of fn and fp Signals



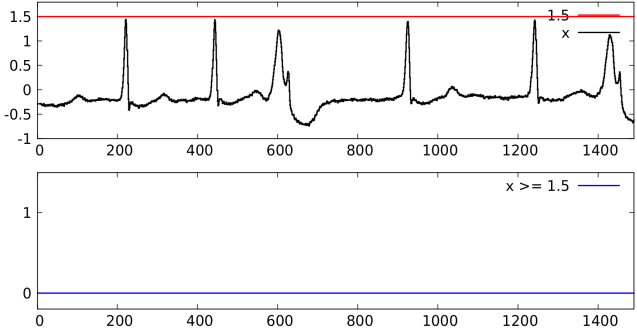
- How to get a (monotonic) measure from the fp and fn signals?
- Why monotonic?

Outline

1. Learning Parameters for STL Specifications
   - Background: Signal Temporal Logic (STL)
   - STL for Accurate Pattern Prediction
   - Monotonic Formulae
   - Algorithm
   - Experiments

Monotonic Formulae

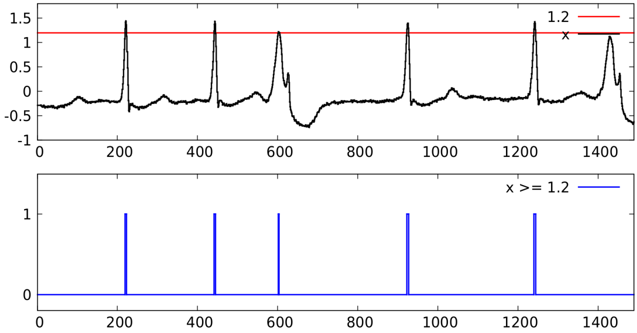
Example monotonic (w.r.t. p) formula: $(x \geq -p)$ $(p = -1.5)$, (No $\texttt{fp}$)



Number of true intervals: 0
Cumulative length of true intervals: 0

Monotonic Formulae

Example monotonic (w.r.t. p) formula: $(x \geq -p) \quad (p = -1.2)$

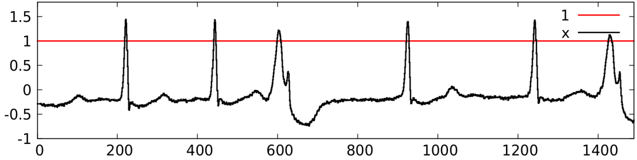


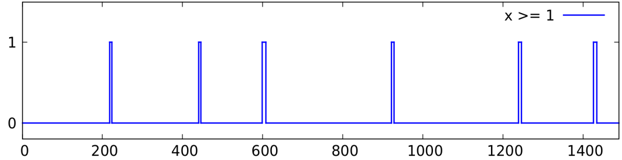
Number of true intervals: 5

Cumulative length of true intervals: 19

Monotonic Formulae

Example monotonic (w.r.t. p) formula: $(x \geq -p) \quad (p = -1.0)$



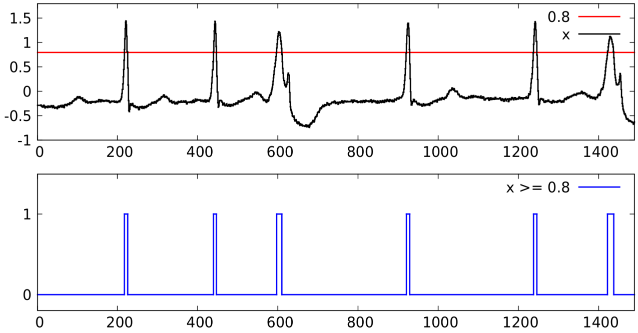


Number of true intervals: 6

Cumulative length of true intervals: 43

Monotonic Formulae

Example monotonic (w.r.t. p) formula: $(x \geq -p)$ $(p = -0.8)$

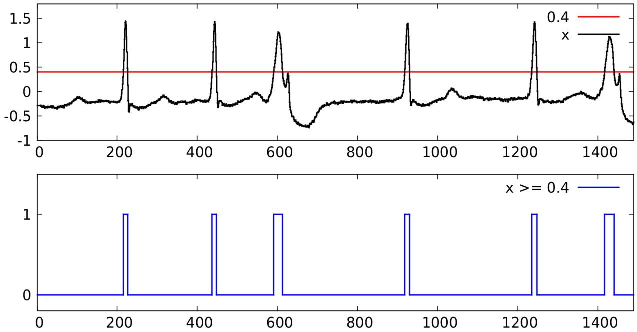


Number of true intervals: 6

Cumulative length of true intervals: 60

Monotonic Formulae

Example monotonic (w.r.t. p) formula: $(x \geq -p) \qquad (p = -0.4)$

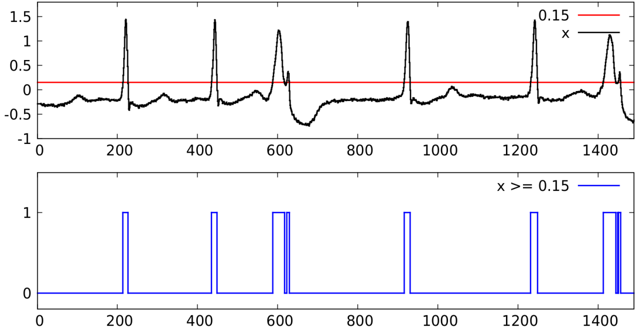


Number of true intervals: 6

Cumulative length of true intervals: 93

Monotonic Formulae

Example monotonic (w.r.t. p) formula: $(x \geq -p) \qquad (p = -0.15)$

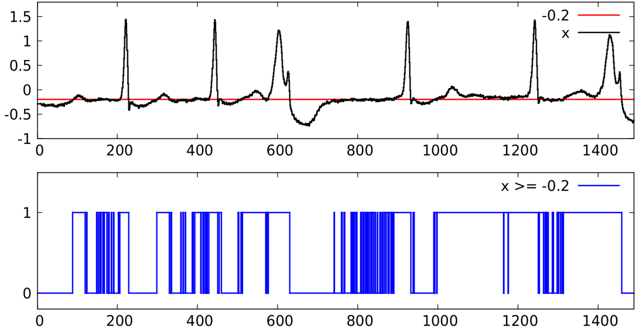


Number of true intervals: 9

Cumulative length of true intervals: 133

Monotonic Formulae

Example monotonic (w.r.t. p) formula: $(x \geq -p)$ $\qquad (p = 0.2)$

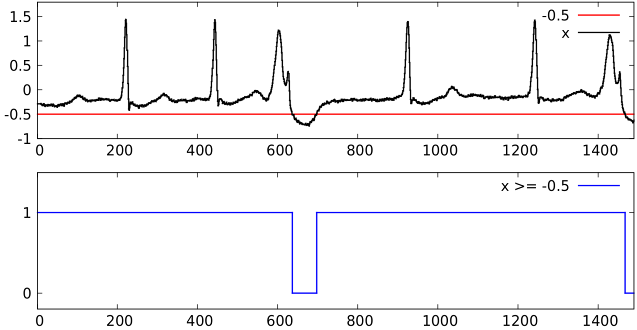


Number of true intervals: 66

Cumulative length of true intervals: 817

Monotonic Formulae

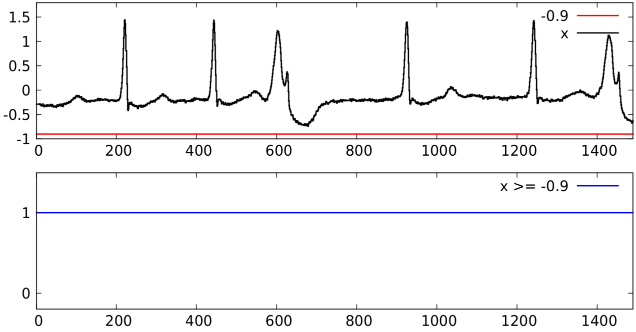
Example monotonic (w.r.t. p) formula: $(x \geq -p)$ $(p = 0.5)$



Number of true intervals: 2
Cumulative length of true intervals: 1407

Monotonic Formulae

Example monotonic (w.r.t. p) formula: $(x \geq -p)$ $(p = 0.9)$(No `fn`)



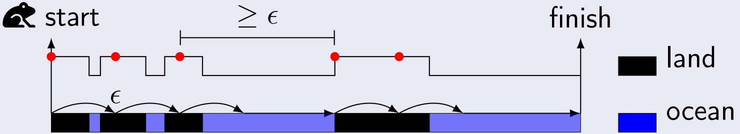
- $p \nearrow$: `fn` $\searrow$: Therefore, given bound on `fn` easy to satisfy
- $p \nearrow$: `fp` $\nearrow$: Therefore, given bound on `fp` difficult to satisfy
- Formula monotonic then `fn` and `fp` also monotonic (w.r.t. p)
- Number of true intervals in Boolean signals is not monotonic (w.r.t. p)

Introducing ϵ-count

- For a Boolean signal w, the maximum number of ϵ-separated points that can be contained in true zone is ϵ-count (monotonic)
- Close to the notion of ϵ-capacity by [Kolmogorov and Tikhomirov, 1959]

Frog analogy for our greedy algorithm

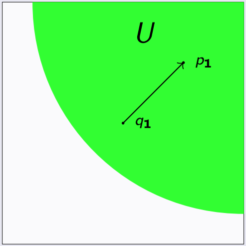
- A primordial one legged frog. It can jump by exactly ϵ when on land
- It can swim any distance in water
- The number of footsteps it leaves is ϵ-count ($= 5$ below)



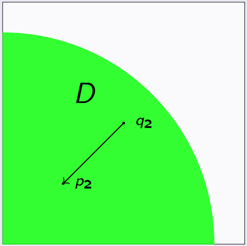
Upset, Downset and Intersection

Subsets of parameter space

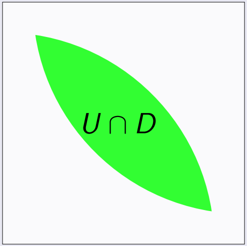
- $\{p \mid$ ϵ-count of false negatives $\leq \texttt{f}_-$ $\}$ is an Upset (upward closed)
- $\{p \mid$ ϵ-count of false positives $\leq \texttt{f}_+$ $\}$ is a Downset (downward closed)
- Intersection: Both the above conditions



(a) Upset: $q_1 \in U \wedge q_1 \leq p_1 \rightarrow p_1 \in U$



(b) Downset: $q_2 \in D \wedge p_2 \leq q_2 \rightarrow p_2 \in D$



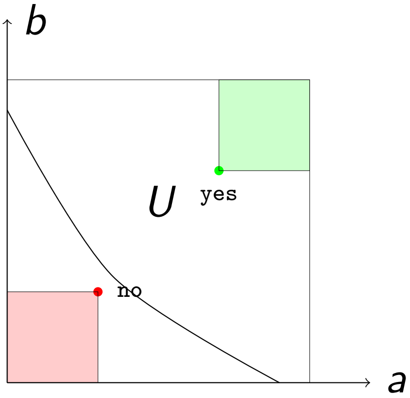
(c) Intersection

Outline

1. Learning Parameters for STL Specifications
   - Background: Signal Temporal Logic (STL)
   - STL for Accurate Pattern Prediction
   - Monotonic Formulae
   - Algorithm
   - Experiments

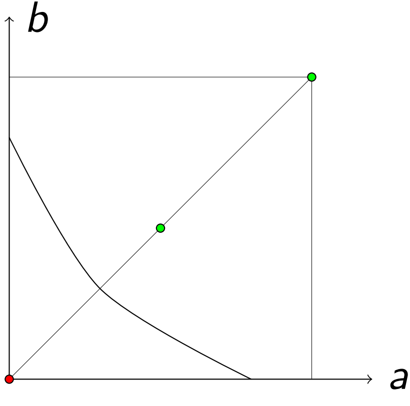
Learning an Upset Via Membership Queries [ParetoLib: Basset, Bakhirkin, Maler, Requeno 2019]

Infer from queries



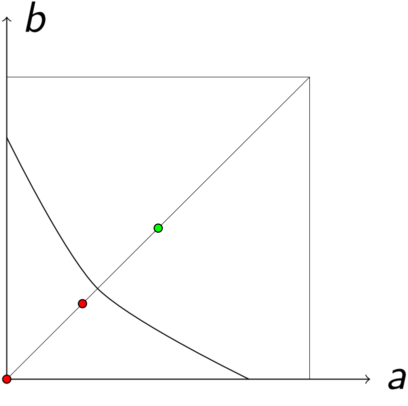
Learning an Upset Via Membership Queries [ParetoLib: Basset, Bakhirkin, Maler, Requeno 2019]

- Binary search on the diagonal using queries



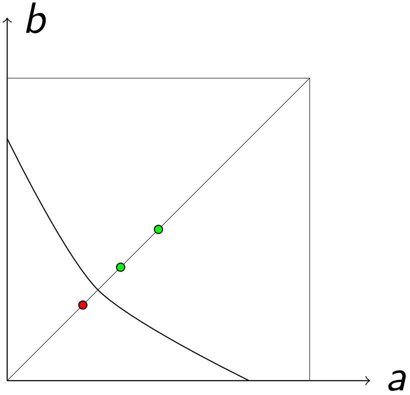
Learning an Upset Via Membership Queries [ParetoLib: Basset, Bakhirkin, Maler, Requeno 2019]

- Binary search on the diagonal using queries



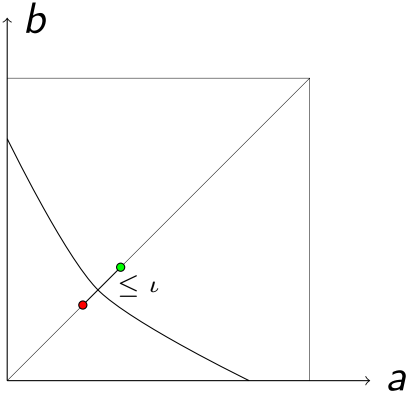
Learning an Upset Via Membership Queries [ParetoLib: Basset, Bakhirkin, Maler, Requeno 2019]

- Binary search on the diagonal using queries



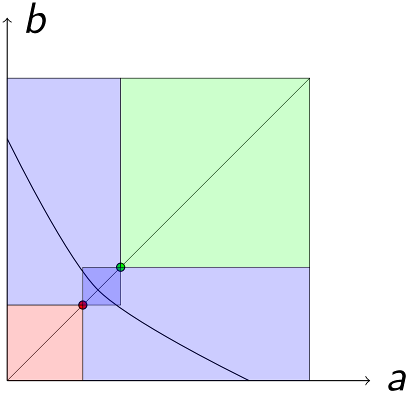
Learning an Upset Via Membership Queries [ParetoLib: Basset, Bakhirkin, Maler, Requeno 2019]

- Binary search on the diagonal using queries
- Find interval (length $\leq \iota$) containing the boundary



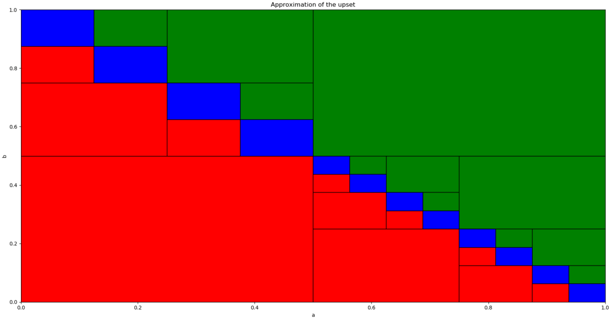
Learning an Upset Via Membership Queries [ParetoLib: Basset, Bakhirkin, Maler, Requeno 2019]

- Binary search on the diagonal using queries
- Find interval (length $\leq \iota$) containing the boundary
- Divide and recurse (on a blue box)



Learning an Upset Via Membership Queries [ParetoLib: Basset, Bakhirkin, Maler, Requeno 2019]

- Approximating upset $(a + b \geq 1)$ after a few iterations
- $\iota = 10^{-5}$ negligible



Learning Intersection of an Upset and a Downset Via Membership Queries
In ParetoLib 2.0

Algorithm Sketch to Process a Box

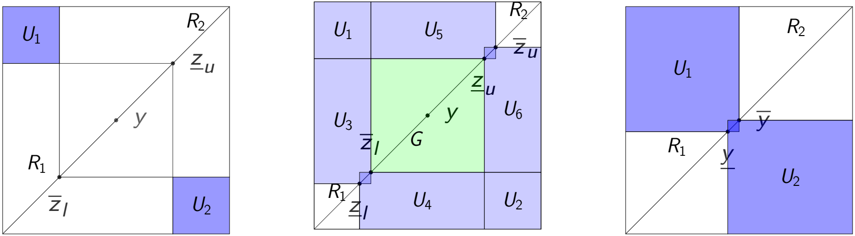
- Binary search on the diagonal to find a point p_c in the intersection
  - $p_c \in U \cap D$ (positive intersection) or
  - $p_c \in U^c \cap D^c$ (negative intersection)
- Binary search two times to find boundary points p_+ and p_-

(a) Negative intersection.

(b) Positive intersection.

Decomposing the Box

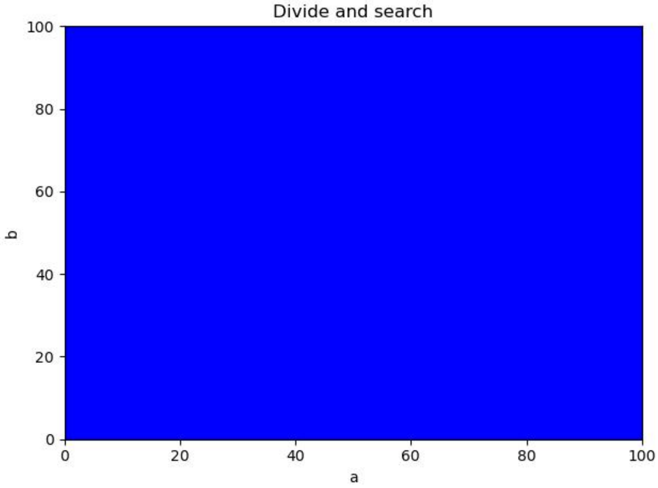
- Deduce the green (positive intersection), white (no solution) and blue (undecided) regions by drawing hyper-rectangles (boxes)
- The number of boxes generated is at most quadratic in dimension of the parameter space
- Add the blue boxes to the processing queue and repeat



(a) Negative intersection (b) Positive intersection (c) No intersection found

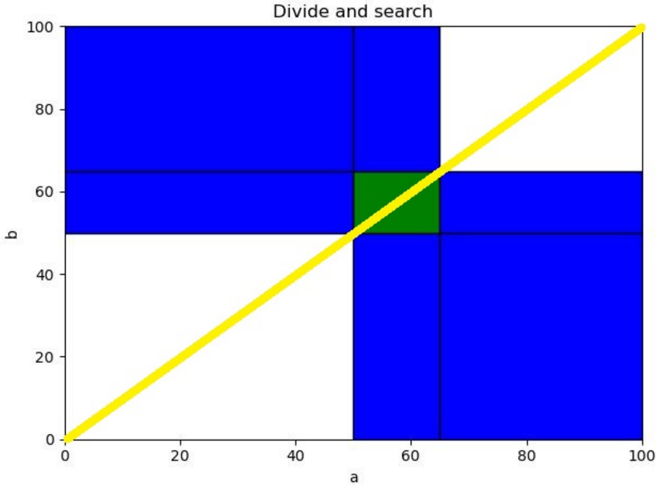
Algorithm Execution (a and b are Parameters)

Explore: $a + b \geq 100 \wedge a + b \leq 130$. Target unexplored volume (V_δ): 10%



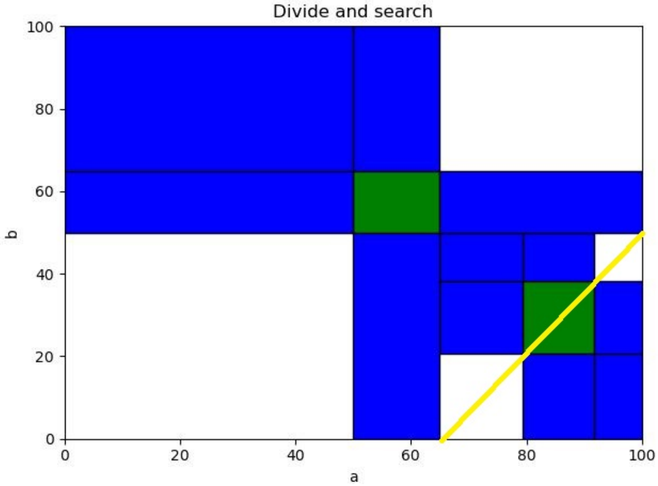
Algorithm Execution (a and b are Parameters)

Explore: $a + b \geq 100 \wedge a + b \leq 130$. Target unexplored volume (V_δ): 10%



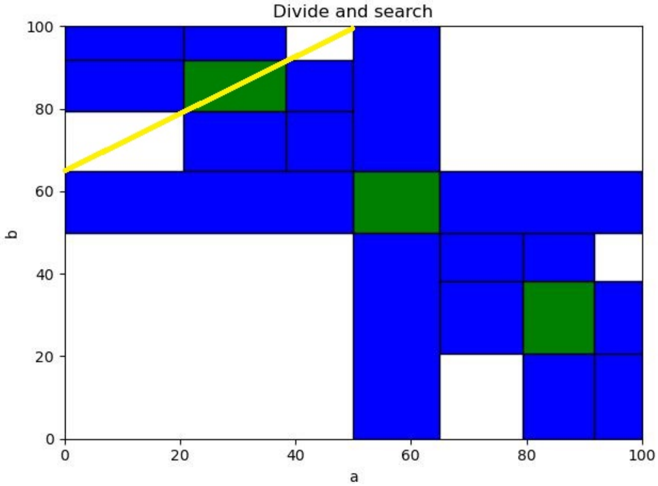
Algorithm Execution (a and b are Parameters)

Explore: $a + b \geq 100 \wedge a + b \leq 130$. Target unexplored volume (V_δ): 10%



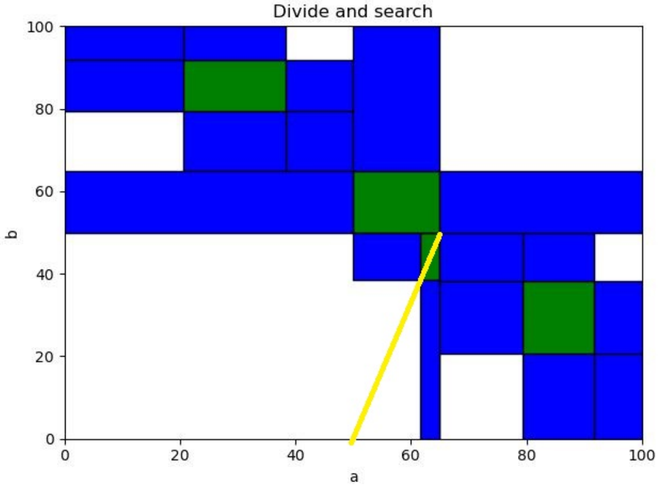
Algorithm Execution (a and b are Parameters)

Explore: $a + b \geq 100 \wedge a + b \leq 130$. Target unexplored volume (V_δ): 10%



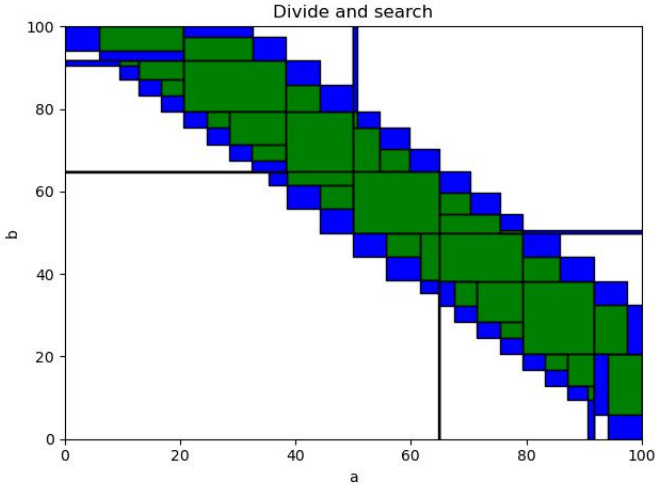
Algorithm Execution (a and b are Parameters)

Explore: $a + b \geq 100 \wedge a + b \leq 130$. Target unexplored volume (V_δ): 10%



Algorithm Execution (a and b are Parameters)

Explore: $a + b \geq 100 \wedge a + b \leq 130$. Target unexplored volume (V_δ): 10%



Outline

1. Learning Parameters for STL Specifications
   - Background: Signal Temporal Logic (STL)
   - STL for Accurate Pattern Prediction
   - Monotonic Formulae
   - Algorithm
   - Experiments

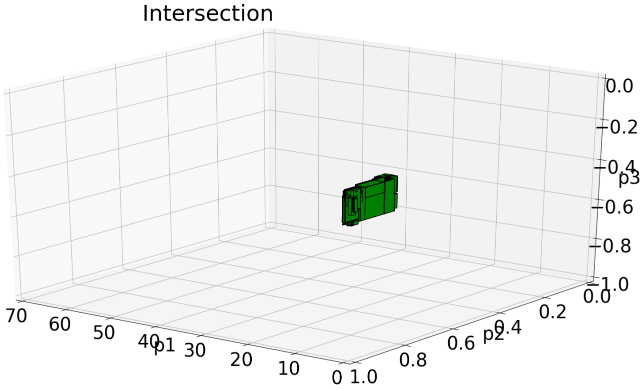
Experiments

- ECG signals taken from MIT-BIH Arrhythmia Database of Physionet [Goldberger et al., 2000, Moody and Mark, 2001]
- Give solution set of parameters with the best possible trade-off between false positives and false negatives

Characterization of ECG Pulses

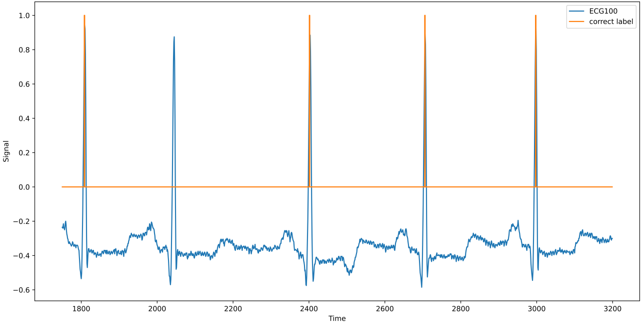
- $\Psi^{ch}_{(p_1,p_2,p_3)} := ((\mathrm{Max}_{[-\mathrm{c},-\mathrm{p}_1]}\, s - \mathrm{Min}_{[-\mathrm{c},-\mathrm{p}_1]}\, s) \leq p_2) \wedge ((\mathrm{Max}_{[-\mathrm{p}_1,\mathrm{p}_1]}\, s) \geq -p_3) \wedge ((\mathrm{Max}_{[\mathrm{p}_1,\mathrm{c}]}\, s - \mathrm{Min}_{[\mathrm{p}_1,\mathrm{c}]}\, s) \leq p_2)$
- For ECG-221, no false negatives (fn) and a single false positive (fp) out of 2462 annotations
- For ECG-100, neither the number of fp nor fn can go below 30 out of 2274 annotations
- Reason: Expressed only the shape of the heart pulses not their rhythm

3D Intersection/Solution Set (ECG 221) (zero false negatives and one false positive)



ECG 100: Atrial Premature Beat

- $\Psi^{ch_b}_{(p_1,p_2,p_3,p_4)} := \Psi^{ch}_{(p_1,p_2,p_3)} \wedge F_{[0,p_4]} \Psi^{ch}_{(p_1,p_2,p_3)}$
- Above peak detector has 3 false positives and 1 false negative



Outline

2. Parametric Timed Regular Expressions
   - Pattern Matching
   - Parametric Signal Regular Expressions (PSRE)
   - Parametric Timed Regular Expressions (Event-Based)
   - Parametric Identification of PSRE

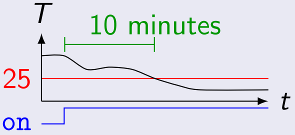
Matching Words and Timed Pattern Matching

Classical pattern matching

Search for "cow" in "The cow gives milk."

Timed pattern matching [Ulus et al., 2014]

Find time intervals (t, t'): $\uparrow \texttt{on} \cdot \langle T \geq 25 \rangle_{[10,\infty)}$ (air-conditioning malfunction)



Parametric Timed Pattern Matching (PTPM)

- Done for event-based Parametric Timed Automata (PTA) by [Waga and André, 2019]
- In this thesis, for three kinds of parametric TRE

Three Types of Parametric TRE

Parametric Signal Regular Expressions (PSRE)

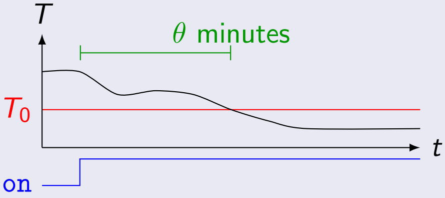
Parametric Timed Regular Expressions (PTRE) (event-based)

Parametric Event-Bounded Timed Regular Expressions (PE-TRE)

Three Types of Parametric TRE

Parametric Signal Regular Expressions (PSRE) extends SRE [Bakhirkin et al., 2017]

- $\langle T \geq T_0 \rangle_{[\theta,\infty)}$



Parametric Timed Regular Expressions (PTRE) (event-based)

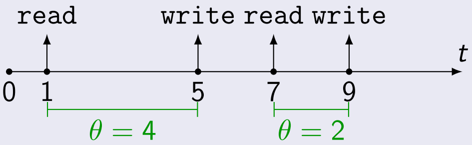
Parametric Event-Bounded Timed Regular Expressions (PE-TRE)

Three Types of Parametric TRE

Parametric Signal Regular Expressions (PSRE)

Parametric Timed Regular Expressions (PTRE) (event-based) extends TRE [Asarin et al., 2002]

- $\underline{\texttt{read}} \cdot \langle \underline{\texttt{write}} \rangle_{[\theta,\theta]}$



Parametric Event-Bounded Timed Regular Expressions (PE-TRE)

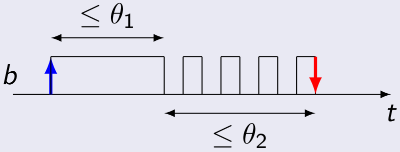
Three Types of Parametric TRE

Parametric Signal Regular Expressions (PSRE)

Parametric Timed Regular Expressions (PTRE) (event-based)

Parametric Event-Bounded Timed Regular Expressions (PE-TRE) extends E-TRE [Ferrère et al., 2015]

- Rising and falling edges as events
- $\uparrow b \cdot \langle b \rangle_{[0,\theta_1]} \cdot \langle \neg b \cdot b \rangle^{+}_{[0,\theta_2]} \cdot \downarrow b$



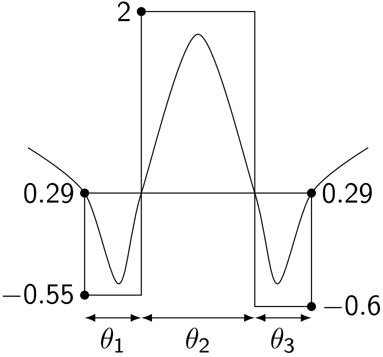
Outline

2. Parametric Timed Regular Expressions
   - Pattern Matching
   - Parametric Signal Regular Expressions (PSRE)
   - Parametric Timed Regular Expressions (Event-Based)
   - Parametric Identification of PSRE

Specification with PSRE: ECG Example

PSRE (with simple concatenation):

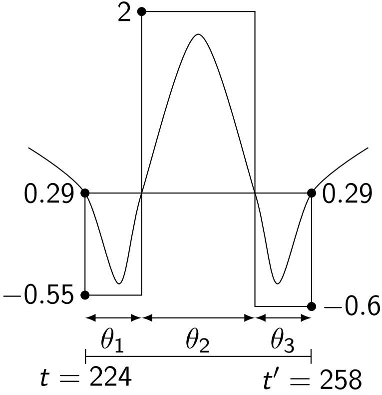
$$\langle -0.55 \leq x \leq 0.29 \rangle_{[\theta_1,\theta_1]} \cdot \langle 0.29 \leq x \leq 2.0 \rangle_{[\theta_2,\theta_2]} \cdot \langle -0.6 \leq x \leq 0.29 \rangle_{[\theta_3,\theta_3]}$$



Specification with PSRE: ECG Example

PSRE (with simple concatenation):

$$\langle -0.55 \leq x \leq 0.29 \rangle_{[\theta_1,\theta_1]} \cdot \langle 0.29 \leq x \leq 2.0 \rangle_{[\theta_2,\theta_2]} \cdot \langle -0.6 \leq x \leq 0.29 \rangle_{[\theta_3,\theta_3]}$$

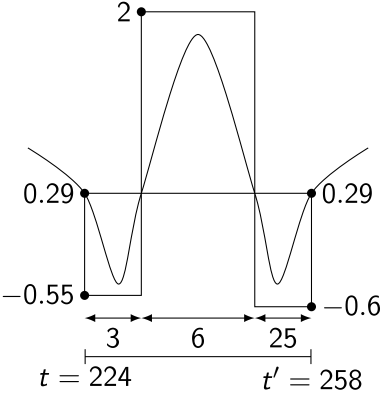


timed match

Specification with PSRE: ECG Example

PSRE (with simple concatenation):

$$\langle -0.55 \leq x \leq 0.29 \rangle_{[\theta_1,\theta_1]} \cdot \langle 0.29 \leq x \leq 2.0 \rangle_{[\theta_2,\theta_2]} \cdot \langle -0.6 \leq x \leq 0.29 \rangle_{[\theta_3,\theta_3]}$$



parametric timed match

Parametric Signal Regular Expressions (PSRE)

PSRE (syntax):

$$x \geq \lambda \mid x \leq \lambda \mid x \geq c \mid x \leq c \mid \langle \phi \rangle_{[a(\overline{\theta}), b(\overline{\theta})]} \mid \phi \cdot \psi \mid \phi^* \mid \phi \wedge \psi \mid \phi \vee \psi$$

- x: signal variable
- $\overline{\theta}$: parameters
- $a(\overline{\theta})$, $b(\overline{\theta})$: affine expressions over $\overline{\theta}$
- $\lambda \in \overline{\theta}$: a parameter
- $c \in \mathbb{R}$: a constant

Parametric match set:

$$\mathcal{M}(w, \phi(\overline{\theta})) = \{(t, t', \overline{\theta}) \mid (w, t, t') \models \phi(\overline{\theta})\}$$

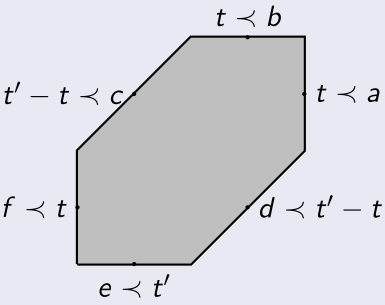
Main aspects

- Describe sequences naturally
- Duration and magnitudes
- Parameters for more expressiveness

Main theorem of timed pattern matching [Ulus et al., 2014]

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

Zone



$$\begin{pmatrix} t \prec a \\ t' \prec b \\ t' - t \prec c \\ d \prec t' - t \\ e \prec t' \\ f \prec t \end{pmatrix}$$

Method for proof and algorithm

Structural induction over φ

Main theorem of parametric timed pattern matching

The **parametric** match-set $\mathcal{M}(\varphi, w)$ is a finite union of **parametric** zones

Parametric zone (z)

$$\begin{pmatrix} t \prec \min(a_1(\overline{\theta}), \ldots, a_{n_1}(\overline{\theta})) \\ t' \prec \min(b_1(\overline{\theta}), \ldots, b_{n_2}(\overline{\theta})) \\ t' - t \prec \min(c_1(\overline{\theta}), \ldots, c_{n_3}(\overline{\theta})) \\ \max(d_1(\overline{\theta}), \ldots, d_{n_4}(\overline{\theta})) \prec t' - t \\ \max(e_1(\overline{\theta}), \ldots, e_{n_5}(\overline{\theta})) \prec t' \\ \max(f_1(\overline{\theta}), \ldots, f_{n_6}(\overline{\theta})) \prec t \end{pmatrix} \wedge \mathrm{C_z}(\overline{\theta}) \text{ where } \mathrm{C_z}(\overline{\theta}) \text{ is a polytope}$$

over $\overline{\theta}$

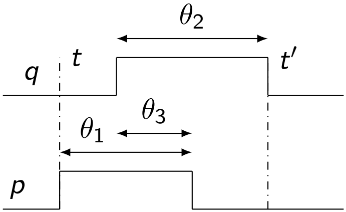
Method for proof and algorithm

Structural induction over φ

Using PSRE Intersection: Example

PSRE:

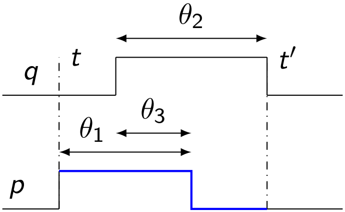
$$(\langle p\rangle_{[\theta_1,\theta_1]}\cdot\neg p)\wedge(\neg q\cdot\langle q\rangle_{[\theta_2,\theta_2]})\wedge(\mathtt{true}\cdot\langle p\wedge q\rangle_{[\theta_3,\theta_3]}\cdot\mathtt{true})$$



Using PSRE Intersection: Example

PSRE:

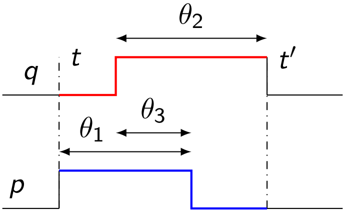
$$(\langle p \rangle_{[\theta_1,\theta_1]} \cdot \neg p) \wedge (\neg q \cdot \langle q \rangle_{[\theta_2,\theta_2]}) \wedge (\mathtt{true} \cdot \langle p \wedge q \rangle_{[\theta_3,\theta_3]} \cdot \mathtt{true})$$



Using PSRE Intersection: Example

PSRE:

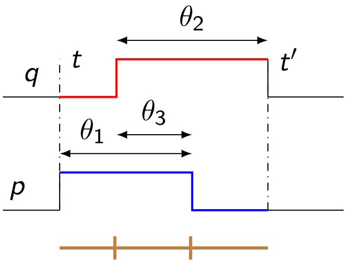
$$(\langle p\rangle_{[\theta_1,\theta_1]} \cdot \neg p) \wedge (\neg q \cdot \langle q\rangle_{[\theta_2,\theta_2]}) \wedge (\mathtt{true} \cdot \langle p \wedge q\rangle_{[\theta_3,\theta_3]} \cdot \mathtt{true})$$



Using PSRE Intersection: Example

PSRE:

$$(\langle p\rangle_{[\theta_1,\theta_1]} \cdot \neg p) \wedge (\neg q \cdot \langle q\rangle_{[\theta_2,\theta_2]}) \wedge (\texttt{true} \cdot \langle p \wedge q\rangle_{[\theta_3,\theta_3]} \cdot \texttt{true})$$

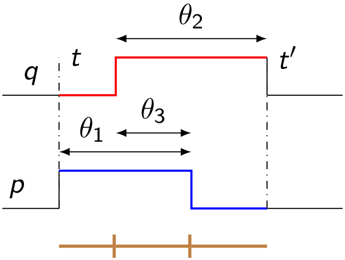


Using PSRE Intersection: Example

PSRE:

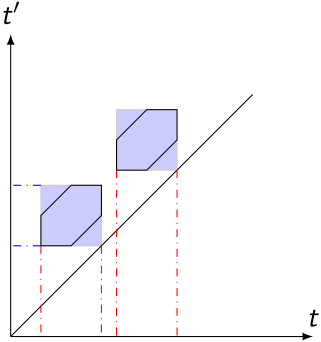
$$(\langle p\rangle_{[\theta_1,\theta_1]}\cdot\neg p)\wedge(\neg q\cdot\langle q\rangle_{[\theta_2,\theta_2]})\wedge(\texttt{true}\cdot\langle p\wedge q\rangle_{[\theta_3,\theta_3]}\cdot\texttt{true})$$

A parametric zone in parametric match-set: $(t'-\theta_2=3)\wedge(t+\theta_1=7)\wedge(\theta_3\leq 4)\wedge(t'\leq 10)\wedge(t\leq 3)\wedge(\theta_3\geq 1)\wedge(t\geq 0)\wedge(t'\geq 7)$



Algorithms for Intersection and Concatenation: Plane-Sweep

- Intersect and concatenate formulae: intersect and sequential composition of zones respectively
- All possible combinations: quadratic
- Handle parametric zones by ordering them along the time dimensions



- Project on time dimensions
- Infer (~~compute~~) empty intersection and sequential composition
- Linear number of polytope operations for usual cases

Outline

2. Parametric Timed Regular Expressions
   - Pattern Matching
   - Parametric Signal Regular Expressions (PSRE)
   - Parametric Timed Regular Expressions (Event-Based)
   - Parametric Identification of PSRE

Parametric Timed Regular Expressions (PTRE) (Event-Based) Syntax

Syntax:

$$\underline{a} \mid \langle \varphi \rangle_I \mid \varphi_1 \cdot \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \varphi^*$$

where $I = [\alpha, \beta]$ and each of α, β is either a non-negative constant or a timing parameter s_i

Σ is the event alphabet

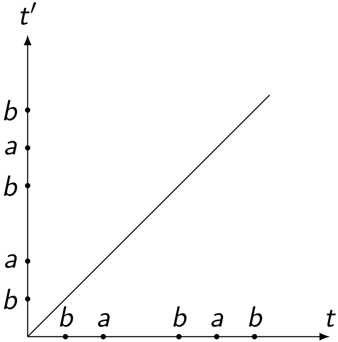
For $a \in \Sigma$, $\underline{a}$ represents an arbitrary passage of time followed by event a

Express

- Sequences of events
- Distribution of events in time and their relations

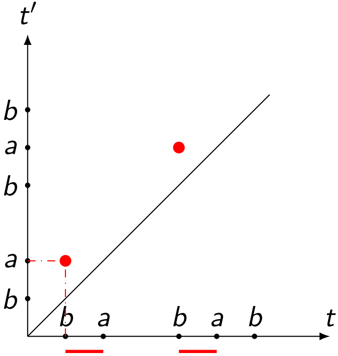
PTRE (Event-Based) Matching Illustration

- The timed word: $(1, b), (2, a), (4, b), (5, a), (6, b)$
- Whole expression: $\langle \underline{a} \cdot \underline{b} \rangle_{[\theta_1, \theta_2]}$
-



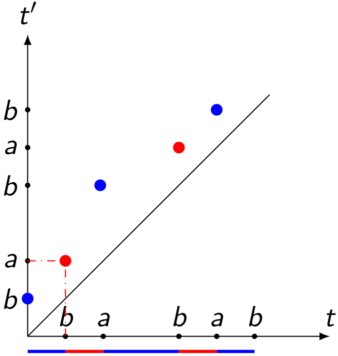
PTRE (Event-Based) Matching Illustration

- The timed word: $(1, b), (2, a), (4, b), (5, a), (6, b)$
- Whole expression: $\langle \underline{a} \cdot \underline{b} \rangle_{[\theta_1, \theta_2]}$
- $\underline{a}$



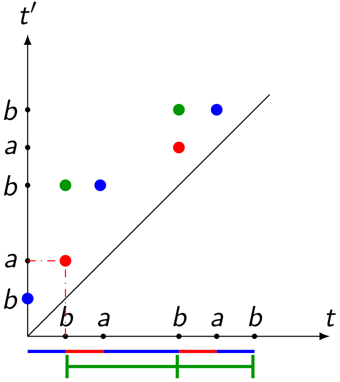
PTRE (Event-Based) Matching Illustration

- The timed word: $(1, b), (2, a), (4, b), (5, a), (6, b)$
- Whole expression: $\langle \underline{a} \cdot \underline{b} \rangle_{[\theta_1, \theta_2]}$
- $\underline{a}$ $\underline{b}$



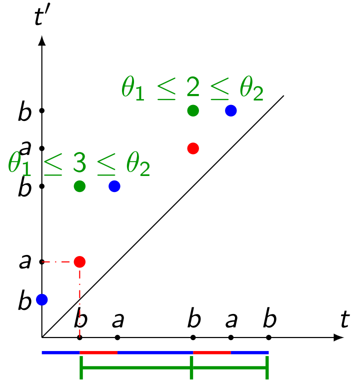
PTRE (Event-Based) Matching Illustration

- The timed word: $(1, b), (2, a), (4, b), (5, a), (6, b)$
- Whole expression: $\langle \underline{a} \cdot \underline{b} \rangle_{[\theta_1, \theta_2]}$
- $\underline{a} \cdot \underline{b}$



PTRE (Event-Based) Matching Illustration

- The timed word: $(1, b), (2, a), (4, b), (5, a), (6, b)$
- Whole expression: $\langle \underline{a} \cdot \underline{b} \rangle_{[\theta_1,\theta_2]}$
- $\langle\ \underline{a}\ \cdot\ \underline{b}\ \rangle_{[\theta_1,\theta_2]}$



Match-Set and Parametric Intervals

Main theorem for PTPM of PTRE (event-based)

For a given timed word the parametric match-set of a PTRE is a finite union of parametric intervals

Parametric interval (y)

$$(t = d_1 \wedge t' = d_2 \wedge \boldsymbol{C_y})$$

where d_1, d_2 are real constants and $\boldsymbol{C_y}$ is a polytope over the parameter space

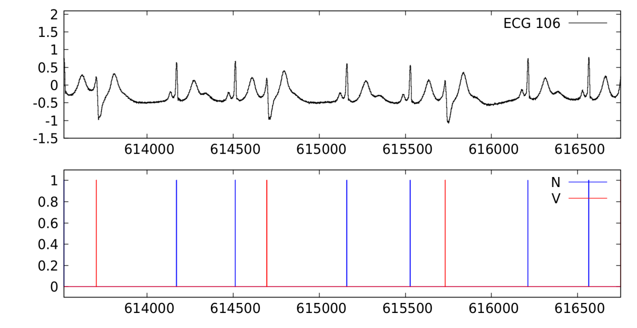
Algorithms for intersection and concatenation

Sort according to the start and end of intervals. Then use binary search

Example: Ventricular Trigeminy

- Repetition of one premature **V**entricular contraction (V) followed by two **N**ormal pulses (N) (previously in [Bartocci et al., 2014])
- $\langle\langle \underline{V} \cdot \underline{N} \cdot \underline{N} \rangle_{[\theta_1,\theta_1]} \cdot (\langle \underline{V} \cdot \underline{N} \cdot \underline{N} \rangle_{[\theta_2,\theta_3]})^{+} \cdot \underline{V} \rangle_{[\theta_4,\theta_4]}$

An example of match (parametric interval)



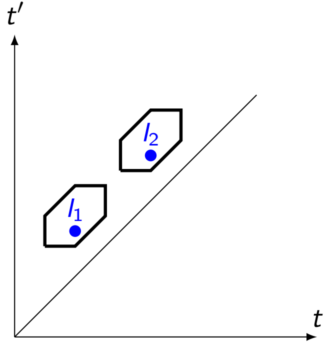
$$(t = 613516) \wedge (t' = 616750) \wedge (\theta_1 = 995) \wedge (0 \leq \theta_2 \leq 1016) \wedge$$
$$(1037 \leq \theta_3) \wedge (\theta_4 = 3234)$$

Outline

2. Parametric Timed Regular Expressions
    - Pattern Matching
    - Parametric Signal Regular Expressions (PSRE)
    - Parametric Timed Regular Expressions (Event-Based)
    - Parametric Identification of PSRE

Parametric Identification of PSRE

- Given a PSRE ψ (defined over parameter vectors u and v)
- Signal w
- A list $\mathcal{I}$ of n intervals $I_1 = [a_1, b_1], \ldots, I_n = [a_n, b_n]$
- Compute the set of parameters that produce a match at each of these n intervals



Parametric Identification of ECG

PSRE for characterizing and identifying ECG:

$$\phi_7 := \langle -\lambda_0 \leq x \leq \lambda_1 \rangle_{[0,\theta_0]} \cdot \langle \lambda_1 \leq x \leq \lambda_0 \rangle_{[0,\theta_1]} \cdot \langle -\lambda_0 \leq x \leq \lambda_1 \rangle_{[0,\theta_0]}$$

- We label the first ten pulses of ECG using ten intervals of 80 time units each
- Parametric identification takes 478s. Inefficient for large data

How to side-step this inefficiency?

- Bottleneck is computing match-set when magnitude parameters (λ) for expressions like $x \leq \lambda$
- Avoid magnitude parameters. Pre-process signals using Booleanizers
- Then detect patterns using expressions with only timing parameters

Conclusions: PSTL

- Theoretical contributions
  - Use of STL formulae as pattern predictors
  - Inaccuracy of prediction represented using false positive and false negative signals
  - ϵ-count to quantify the above signals
  - Intersection algorithm for parametric identification of monotonic predictors using queries
- Implementation of intersection algorithm in ParetoLib
- Application to analysis of ECGs
  - Parametric identification of normal pulse/peak detection in ECGs
  - Classification of ECGs via feature learning

Conclusions: parametric TRE

- Theoretical contributions
  - Three types of parametric TRE suitable for expressing different types of behaviours (in signals and timed words)
  - Algorithms for parametric timed pattern matching of the above
  - Parametric identification of PSRE
- Implemented `parameTRE` tool
- Applications:
  - Analysis of ECG signals
  - Detection of marine traffic rule violation

Conclusions

Not presented today:

- Counting operators for (parametric) STL
- Extension of (parametric) TRE with Min/Max operators
- Concept of “decisive region” in (parametric) timed pattern matching

Future Work

- Monotonic PSRE (timing parameters only?) defined over PSTL Booleanizers (magnitude parameters only?) in ParetoLib
- Diagnostics of TRE and parametric TRE. Use match-set graph

References I

Asarin, E., Caspi, P., and Maler, O. (2002).
Timed regular expressions.
J. ACM, 49(2):172–206.

Bakhirkin, A. and Basset, N. (2019).
Specification and efficient monitoring beyond STL.
In TACAS (2), volume 11428 of Lecture Notes in Computer Science, pages 79–97. Springer.

Bakhirkin, A., Ferrère, T., Maler, O., and Ulus, D. (2017).
On the quantitative semantics of regular expressions over real-valued signals.
In Abate, A. and Geeraerts, G., editors, Formal Modeling and Analysis of Timed Systems - 15th International Conference, FORMATS 2017, Berlin, Germany, September 5-7, 2017, Proceedings, volume 10419 of Lecture Notes in Computer Science, pages 189–206. Springer.

References II

Bartocci, E., Bortolussi, L., and Sanguinetti, G. (2014).
Data-driven statistical learning of temporal logic properties.
In Legay, A. and Bozga, M., editors, Formal Modeling and Analysis of Timed Systems - 12th International Conference, FORMATS 2014, Florence, Italy, September 8-10, 2014. Proceedings, volume 8711 of Lecture Notes in Computer Science, pages 23–37. Springer.

Ferrère, T., Maler, O., Ničković, D., and Ulus, D. (2015).
Measuring with timed patterns.
In CAV (2), volume 9207 of Lecture Notes in Computer Science, pages 322–337. Springer.

References III

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000).
PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals.
Circulation, 101(23):e215–e220.

Kolmogorov, A. N. and Tikhomirov, V. M. (1959).
ε-entropy and ε-capacity of sets in function spaces.
Uspekhi Matematicheskikh Nauk, 14(2(86)):386.

Maler, O. and Ničković, D. (2004).
Monitoring temporal properties of continuous signals.
In Lakhnech, Y. and Yovine, S., editors, Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Joint International Conferences on Formal Modelling and Analysis of Timed Systems, FORMATS 2004 and Formal Techniques in Real-Time and Fault-Tolerant Systems, FTRTFT 2004, Grenoble, France, September 22-24, 2004, Proceedings, volume 3253 of Lecture Notes in Computer Science, pages 152–166. Springer.

Moody, G. B. and Mark, R. G. (2001).
The impact of the MIT-BIH arrhythmia database.
IEEE Engineering in Medicine and Biology Magazine, 20(3):45–50.

References V

Ulus, D., Ferrère, T., Asarin, E., and Maler, O. (2014).
Timed pattern matching.
In FORMATS, volume 8711 of Lecture Notes in Computer Science, pages 222–236. Springer.

Waga, M. and André, É. (2019).
Online parametric timed pattern matching with automata-based skipping.
In Badger, J. M. and Rozier, K. Y., editors, NASA Formal Methods - 11th International Symposium, NFM 2019, Houston, TX, USA, May 7-9, 2019, Proceedings, volume 11460 of Lecture Notes in Computer Science, pages 371–389. Springer.