Summary of the papers for the report

Contents

1 "On temporal logic and signal processing"

Object: Presentation of a new temporal logic for REAL-VALUED SIGNALS with a checking framework

STL = temporal logic proprieties + frequency-domain properties

Temporal Logic puts the focus on the ongoing input-output behavior of reactive systems, rather than the final output.

Reactive systems = viewed by designers as networks of transducers (block diagrams) communicating continuously via signals.

Signal = function from Time ($\mathbb{T}$) to Real numbers ($\mathbb{R}$)

TL provides a framework to write in a formal way specifications that the system under design should satisfy. It has been used to specify proprieties of real-valued signals defined over dense time domains.

STL allows designers to speak of proprieties relates to the order of discrete events and the temporal distance between them.

Event = change in the satisfaction of some predicate over the real variables.

Traditional measures for signals are more continuous and event-free: averages/discounted integrals of some variables over time.

One of the impediment to STL was its purely time-domain nature that does not match with the frequency-domain analysis. Frequency-domain analysis is based on Fourier spectrum of the signal, that leave aside the time at which events occur. This is done to deal with **noise**.

Noise = random perturbations in the designed signal, in the frequency domain dealt with filtering techniques. It populates range of frequencies different from the designed signal, and therefore we keep only the latter, then we reduce the noise (*separation of the source*).

These kinds of operations are time-invariant, since the Fourier transform is defined over an unbounded time-interval, i.e. $t \in (-\infty, +\infty)$. If we have bounded intervals the Fourier transform is not so useful, since its definition aggregates for each frequency all values along the duration of the signal.

Therefore, the aim is to search for proper **time-frequency analysis techniques**. In this paper the authors try to do so by proposing a unified **logical formalism** for expressing hybrid proprieties of signals.

1.1 SIGNAL TEMPORAL LOGIC

Consider a general framework of data-flow systems and add the temporal operators (until and since) as a special type of signal transducers.
Style of presentation of the semantics using temporal testers, applied to translate STL monitoring.
A formula φ is a network of signal operator (transducers), starting with a raw signal x (sequences of atomic propositions in the discrete case), and cumulating in a top-level signal φ, whose vale at t represents the satisfaction of the top-level formula at time t:

$$\varphi[t] = 1 \Leftrightarrow (x,t) \vDash \varphi$$

Each sub-formula of the form $\varphi = f(\psi_1, \psi_2)$ is associated with a signal transducer f that takes as input the satisfaction signals of ψ_1 and ψ_2.
The apparatus of monitoring the satisfaction of a formula by a signal can be viewed as a network of operators working on

- **numerical signals**: raw signals and those obtained by numerical operations on them,
- **Boolean signals**: satisfaction signals of sub-formulae.

Assume that the signals are defined as functions from time $\mathbb{T}$ to some domain $\mathcal{D}$, the range $\mathbb{T}$ can be finite $[0,r]$, infinite $[0,+\infty)$ or bi-infinite $(-\infty,+\infty)$.

Definition. *A signal operator is a mapping $f : (\mathbb{T} \to \mathcal{D}_1) \to (\mathbb{T} \to \mathcal{D}_2)$, where $\mathcal{D}_1$ and $\mathcal{D}_2$ are respectively domains of input and output signals.*

All the operators are computable so that given some representation of a signal x, we can produce a representation of the output signal $f(x)$.

Definition. *Let f be a signal operator and let $y = f(x)$. We say that f is:*

- ***memoryless** if it is lifting to signals of a function $f : \mathcal{D}_1 \to \mathcal{D}_2$, that is, $\forall t\ y[t] = f(x[t])$*
- ***casual** if for every t, $y[t]$ is computed based only on at most $x[0], \ldots, x[t]$,*
- ***acasual** otherwise.*

Casual operators: past operator, back-shifts or integral over temporal window that extend them. They can be monitored online, and it is important for real systems.
Acasual operators: future operator and operators and the ones depending on values in the temporal windows that extend beyond the current time.

Definition. *Let φ_1 and φ_2 be two Boolean signals, and let $a, b \in \mathbb{R}$ such that $a \leq b$. Then $\psi = \varphi_1 \,\mathcal{U}_{[a,b]}\, \varphi_2$ are signals satisfying:*

$$(x,t) \vDash \psi \Leftrightarrow \exists t' \in [a+t, b+t], (x,t') \vDash \varphi_2 \ \wedge\ \forall t'' \in [t,t'](x,t'') \vDash \varphi_1[t'']$$

Since the Boolean operators '$\vee$' and '$\wedge$' are definable with min and max. We can rewrite $\psi[t]$ as follows:

$$\psi[t] = \max_{t' \in [a+t, b+t]} \min(\min_{t'' \in [t,t']} \varphi_1[t''], \varphi_2[t'])$$

We have the derived temporal operators $\Diamond_{[a,b]}\, \varphi = \top\, \mathcal{U}_{a,b}\, \varphi$ and $\Box_{[a,b]}\, \varphi = \neg \Diamond_{[a,b]} \neg \varphi$, which means that:

$$(x,t) \vDash \Diamond_{[a,b]}\, \varphi \quad \Leftrightarrow \quad \exists t' \in [a+t, b+t], (x,t') \vDash \varphi$$
$$(x,t) \vDash \Box_{[a,b]}\, \varphi \quad \Leftrightarrow \quad \forall t' \in [a+t, b+t], (x,t') \vDash \varphi$$

If the until operator has no bounds, then it means that it is defined in the interval $[0, \infty)$, and it does not put any metric constraints on the future occurrence of the second argument.
Assume that $x_1, \ldots, x_n$ a set of variables and F is a family of signals operators:

- Arithmetical and Logical operators: $+, \cdot, \min, \max, \wedge, \vee, = and <$;
- Useful mathematical operators as integrals, convolution, etc.;
- Until operator and its derived operators.

Then we can define as follows the syntax of STL :

Definition. *A formula φ is a formula of* STL *iff:*

- *φ is any variable x_i or any rational constant c,*
- *for any φ if it is a formula, then any $f(\varphi)$, where $f \in F$ is a binary operator,*
- *for any φ_1 and φ_2 if they are formulae, then any $f(\varphi_1, \varphi_2)$, where $f \in F$ is a binary operator.*

The semantics of a STL formula φ relative to a raw signal $x = (x_1, \ldots, x_m)$ is immediate since for the atomic formulae the semantics is the signal itself or a constant signal.
For the binary or monodic operators is obtained by applying them to the semantics of the input formulae.
The authors quote a monitoring algorithm for the satisfaction of a formula by computing the satisfaction of the signals of all sub-formulae, presented in [MN04] and [MNP08]. This algorithm works with the tableaux semantics.
[NM07]: algorithm that solves various problems in finite durations.
[DM10]: presents a quantitative semantics that return as output positive or negative numbers indicating how robustly the propriety is satisfied.

1.2 FREQUENCY ANALYSIS

A signal can be transformed into an alternative representation of a weighted sum of basic elementary signals, namely sinusoids of different frequencies and phases. The signal is transformed from a time-domain representation $\mathbb{T} \to \mathcal{D}$ to a function $\hat{x}$ mapping frequencies to their coefficients. For instance, eliminating the high-frequency components can help reduce noise.
The Fourier transform is a key mathematical operation used for this type of processing. It converts the signal into a spectrum of frequencies, offering a continuous representation of the signal's frequency content. Conversely, the inverse Fourier transform reconstructs the original signal from its frequency spectrum.
The Fourier transform has practical applications in various fields, such as noise reduction, audio processing, and image analysis. It allows operations to be performed directly on the frequency components, making it easier to analyze and manipulate signals effectively. However, working with time-localized effects or specific intervals can be challenging with traditional Fourier methods, motivating the development of alternative techniques that combine time and frequency localization.

1.3 COMBINING THE TWO APPROACHES

The **Short-Time Fourier transform (**STFT**)** combines time-frequency analysis by replacing the function used in classical Fourier transform with a function in two parameters, time and frequency. This time of transform the usual transform parametrized on ω multiplying it with the window function, filtering the values of x outside a neighborhood of τ forcing it to be 0. This process can be done with different window functions like the rectangular or more complex ones that that can satisfy the normalization propriety of the Fourier transform, like the Hanning or Gaussian window.
Given a window function, the new STFT is defined as the product of the Fourier transform of the signal and the window function. Therefore, we have that the STFT of a signal x in (ω, τ) defines a two-dimensional spectrum $\{c_{\omega,\tau} : (\omega, \tau) \in \mathbb{R}^2\}$. This function results invertible and can be used to reconstruct the original signal. There are limitations on the trade-off between the precision in time and the precision in frequency, given by the Heisenberg uncertainty principle.

1.3.1 Time-frequency predicates

The STFT of a signal x defines a two-dimensional operator taking time and frequency as arguments. If we consider the frequency as a parameter, and we can define a new family of operators that is the projection of the L-spectrogram of x on frequency ω, i.e. $\{f_{L,\omega}\}$, such that:

$$y = f_{L,\omega}(x) \Leftrightarrow y[t] = \hat{x}_L(\omega, t).$$

Therefore, we can obtain a new time-frequency logic (TFL) adding the operators of the family $\{f_{L,\omega}\}$ to STL . A spectral signal $y = f_{L,\omega}(x)$ can be part of a TFL formula as an argument of a formula. For the monitoring we can use the same process defined for STL , except that we must process before the raw signal x to get the spectrogram, from which the spectral signal y is obtained. This can be done before the monitoring process starts or be integrated in an online procedure. To deduct the frequency at a given time we need the spectrogram matching between the frequency and the time of a raw signal, i.e. that $|c_{\omega,\tau}|$ be non-zero iff x contains a component of the frequency ω at the time τ.
This mapping cannot be obtained for multiple reasons:

- low frequencies require an amount of time that is larger than the corresponding periods to be detectable,
- there are technical limitations on the continuous detectability,
- there is a fundamental limitation related to the uncertainty principle of Heisenberg.

To choose the window is given a practical guide, in order to limit the uncertainty. Then in the rest of the paper is given a practice example in music and how we can detect precise type of melodies given a certain formula from a raw signal.

1.4 CONCLUSION

Presentation of a TFL that is able to represent formally time and frequency proprieties of signals. Using temporal operators can lead to different applications in terms of representation of music signals. But the music is not the only application, TFL can help unify modeling and analysis across different engineering disciplines.

Other extensions include the use of the more versatile Wavelet Transform for time-frequency analysis and the extension of the logic to spatially extended phenomena such as wave propagation, that as a lot of use in the field of medical sciences. The spectrogram is a two-dimensional entity indexed by both time and frequency, and that TFL is currently biased toward time. It would be interesting to explore a specification formalism that can alternate more freely between temporal, frequential and spatial operators.

2 "Monitoring temporal properties of continuous signals"

Object: Presentation of a temporal logic which is a bounded interval of the real-time logic `MITL` and from this present an automatic monitoring that can check if a given signal of bounded length and finite variability satisfies the propriety.

Temporal logic is a formalism used to describe the behavior of discrete systems. The algorithmic verification for a formula in the logic consists of checking whether all state-event sequences generated by a system S satisfy or not a formula φ, i.e. means deciding the language inclusion $[\![S]\!] \subseteq [\![\varphi]\!]$.

For systems outside the automatic verification tools, simulation is the preferred verification method. Authors suggested that the component of verification can be exported toward simulation through propriety monitors.

In monitoring one performs each time a much simpler membership test $\psi \in [\![\varphi]\!]$ on an individual simulation trace $\psi \in [\![S]\!]$ and the responsibility for exhaustive coverage is delegated to the test generation procedure. The essence of this approach is that can be automatized and is much more reliable than the manual checking.

Monitoring is restricted to finite trace. One thread of monitoring research attempts to redefine the semantics of temporal formulae on finite runs, but to avoid the problem we can consider bounded time modalities which we can interpret naturally over finite traces.

Main contribution → Definition of temporal logic for specifying proprieties of dense-time real-valued signals and the automatic generation of propriety monitors for this language.

This in order to improve validation methodology for continuous and hybrid systems, that have the following natural models

Continuous Systems	**Hybrid Systems**
Differential Equations	Automata + Differential Equations
	Hybrid Automata

The exhaustive verification is impossible due to **undecidability**. Numerical simulation is the common method to validate such systems. Some monitoring forms exist in certain numerical simulations, but they are primitive and cannot support properly temporal modalities.

2.1 SIGNALS AND TEMPORAL LOGICS

2.1.1 Signals

Let $\mathbb{T}$ be a time domain and the set $\mathbb{R}_0^+$ the set of non-negative real numbers. A finite length **signal** s over a domain $\mathcal{D}$ is a partial function $s : \mathbb{T} \to \mathcal{D}$, whose domain of definition is the interval $I = [0, r)$, $r \in \mathbb{Q}_0^+$, therefore we have that $r = |s|$ and call it the **length** of the signal.
We define as primitive the notion of $s[t] = \bot$, for every $t \geq |s| = r$. (This holds since it is outside the interval)
To operate with signals in different domains we use the paring and projection operators. Let be $s_1 : \mathbb{T} \to \mathcal{D}_1$,$s_2 : \mathbb{T} \to \mathcal{D}_2$ and $s_3 : \mathbb{T} \to \mathcal{D}_3$ signals. Consider now the function $f : \mathcal{D}_1 \times \mathcal{D}_2 \to \mathcal{D}_3$. Now we can define the operators as follows:

- **Paring**: $s_1 \| s_2 = s_{1,2}$ if $\forall t, s_{1,2}[t] = (s_1[t], s_2[t])$
- **Projection**: is the inverse operator of parsing, such that

$$s_1 = \pi_1(s_{1,2}) \quad \text{and} \quad s_2 = \pi_2(s_{1,2})$$

- **Lifting**: it is defined as the function f to signals, such that:

$$s_3 = f(s_1, s_2) \quad \text{if} \quad \forall t, s_3[t] = f(s_1[t], s_2[t])$$

If s_1 and s_2 differ in length, the convention $f(x, \bot) = f(\bot, x) = \bot$ guarantees that $|s_3| = \min(|s_1|, |s_2|)$. Since we treat Boolean signals $\mathcal{D} = \mathbb{B}$.
An interval covering for an interval $I = [0, r)$ is a sequence of intervals $\mathcal{I} = I_1, I_2, \ldots$, each of them closed only on the left such that $\bigcup I_i = I$ and $I_i \cap I_j = \varnothing$, for every $i \neq j$. Given an interval covering we can say that:

- it is **consistent** with a signal s if for every t, t', $s = [t] = s[t']$;
- a signal s is of **finite variability** if $\mathcal{I}$ is finite, and therefore they are closed under projection and paring, and are also non-Zeno (no infinite number of finite event in a finite time)
- it **refines** another interval covering $\mathcal{I}'$ (i.e. $\mathcal{I} \prec \mathcal{I}'$), if

$$\forall I \in \mathcal{I},\ \exists I' \in \mathcal{I}' \text{ s.t. } I \subseteq I'$$

The **minimal interval covering** consistent with a finite variability signal s is denoted with $\mathcal{I}_s$. the set of positive intervals of s is $\mathcal{I}_s^+ = \{I \in \mathcal{I}_s : s(I) = 1\}$, the set of negatives is it complementary set $\mathcal{I}_s^- = \mathcal{I}_s - \mathcal{I}_s^+$
A **Boolean signal** $s : \mathbb{T} \to \mathbb{B}$ can be represented as a pair $(|s|, \mathcal{I}_s^+)$. It is unitary if $\mathcal{I}_s^+$ is a singleton. We can say that any signal $s = \bigvee_{1 \leq i \leq k} s_i$, where s_i is unitary and the boundaries of each do not intersect.

2.1.2 Real-time Temporal Logic

$\texttt{MITL}_{[a,b]}$ is a fragment of MITL (Metric Interval Temporal Logic), with the modalities restricted to intervals, were $a, b \in \mathbb{Q}_0^+$ and $a < b$. The bounding is required for the monitoring where we observe the behavior of a system in a finite time interval.

$$\textbf{Syntax:} \quad \varphi := p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \mathcal{U}_{[a,b]} \varphi_2$$

Then we have $\Diamond_{[a,b]}\,\varphi$ and $\Box_{[a,b]}\,\varphi$ defined as in the previous paper.
For the **semantics**, we interpret the formulae over n-dimensional Boolean signals. The satisfaction relation $(s,t) \vDash \varphi$, with a signal s satisfying a formula φ, starting from a position t. A signal s satisfies the formula φ iff $(s,0) \vDash \varphi$, is defined as follows:

$$\begin{aligned}
(s,t) \vDash p &\Leftrightarrow \pi_p(s)[t] = \top \\
(s,t) \vDash \neg\varphi &\Leftrightarrow (s,t) \nvDash \varphi \\
(s,t) \vDash \varphi_1 \vee \varphi_2 &\Leftrightarrow (s,t) \vDash \varphi_1 \text{ or } (s,t) \vDash \varphi_2 \\
(s,t) \vDash \varphi_1 \mathcal{U}_{[a,b]} \varphi_2 &\Leftrightarrow \exists t' \in [a+t, b+t]\ (s,t') \vDash \varphi_2 \text{ and } \forall t'' \in [t,t'](s,t'') \vDash \varphi_2 \\
(s,t) \vDash \Diamond_{[a,b]}\,\varphi &\Leftrightarrow \exists t' \in [a+t, b+t]\ (s,t') \vDash \varphi \\
(s,t) \vDash \Box_{[a,b]}\,\varphi &\Leftrightarrow \forall t' \in [a+t, b+t]\ (s,t') \vDash \varphi
\end{aligned}$$

For standard TL the satisfaction of a formula with unbounded modalities can rarely be determined with respect to a finite signal or sequence. The satisfaction of $\Diamond\, p$ and $\Box\, p$ can be detected in a finite time. By bounding the modalities, we solve the ambiguities behind the satisfaction relation, when applied to finite signals. Even for our logic certain signals so we need to restrict to signals of sufficient length. In fact we have that $s \vDash \varphi$ iff $|s| > \|\varphi\|$ We can compute the length as follows:

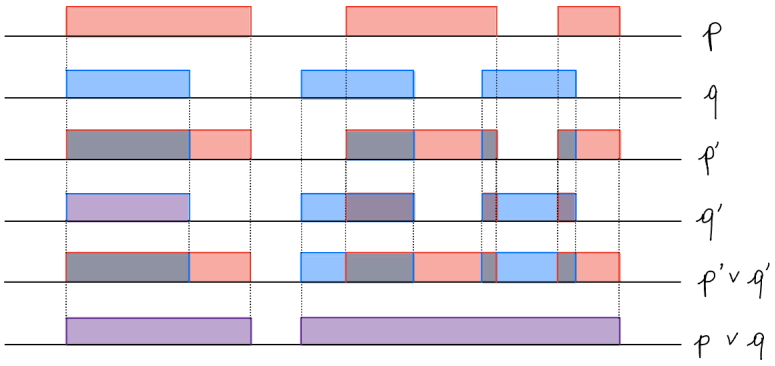
$$\begin{aligned}
\|p\| &= 0 \\
\|\neg\varphi\| &= \|\varphi\| \\
\|\varphi_1 \vee \varphi_2\| &= \max(\|\varphi_1\|, \|\varphi_2\|) \\
\|\varphi_1 \mathcal{U}_{[a,b]} \varphi_2\| &= \max(\|\varphi_1\|, \|\varphi_2\|) + b
\end{aligned}$$

2.2 MONITORING $\mathtt{MITL}_{[a,b]}$ FORMULAE

Procedure for deciding the satisfiablity of a $\mathtt{MITL}_{[a,b]}$ formula. Bottom-up procedure on the parse tree of the formula. Starting from the leaves we construct for every sub-formula ψ a signal s_ψ such that $s_\psi[t] = 1$ iff $(s,t) \vDash \psi$. At the end you have the signal s_φ for the formula whose value at 0 determines the satisfiability.
For future modalities the procedure go backwards, since they talk now as a function of some truth in the future. Is backward since we decompose the formula and the semantics will pass from $[a+t, b+t]$ to t.
For the Boolean operators the computation of a signal for a formula from the signal of its sub-formulae is rather straightforward:

- **Neagtion**: $\mathcal{I}^+_{\neg p} = \mathcal{I}^-_p$
- **Disjunction**: Construct a refined interval covering $\mathcal{I}$ for $p \| q$ and for each $I_i \in \mathcal{I}$, we merge the adjacent positive intervals to obtain $\mathcal{I}^+_\vee$
- **Unitil**: Shift backwards the interval. Let $I = [m, n)$ and $[a, b] \in \mathbb{T}$, the shifting is computed as follows:

$$I \ominus [a,b] = [m-b, n-a)$$



In the picture the computation process for the interval covering the disjunction.

Proposition. *Let p and q be two unitary signals $\mathcal{I}_p^+ = \{I_p\}$ and $\mathcal{I}_q^+ = \{I_q\}$. then the signal $\psi = p\,\mathcal{U}_{[a,b]}\,q$ is a unitary signal satisfying*

$$\mathcal{I}_\psi^+ = \{((I_p \cap I_q) \ominus [a,b]) \cap I_p\}$$

Proof. This follows from the semantics of the operator $\mathcal{U}_{[a,b]}$. Consider a point $t \in \{((I_p \cap I_q) \ominus [a,b]) \cap I_p\}$. By the semantics of the operator we know that there must be a point $t' \in [a+t, b+t]$ where q and p are satisfied and that p is satisfied also at t. Being p unitary, this implies that p holds throughout $[t,t']$. A point $t \notin I_\psi$ will either not have such a point t' or will not satisfy p, and hence not satisfy ψ.

□

Proposition. *Let $p = \bigvee_{i=1}^m p_i$ and $q = \bigvee_{j=1}^n q_j$ be two signals each written as a union of unitary signals. Then*

$$p\,\mathcal{U}_{[a,b]}\,q = \bigvee_{i=1}^{m}\bigvee_{j=1}^{n} p_i\,\mathcal{U}_{[a,b]}\,q_j.$$

Proof. Notice that $p\,\mathcal{U}_{[a,b]}\,(q_1 \vee q_2) = p\,\mathcal{U}_{[a,b]}\,q_1 \vee p\,\mathcal{U}_{[a,b]}\,q_1$. This holds because $q[t]$ is quantified existentially in the semantic definition. The same holds for $(p_1 \vee p_2)\,\mathcal{U}_{[a,b]}\,q$, when the positive intervals are separated. Repeating the reasoning n times for p and m times for q we get the desired result.

□

These imply correctness of the procedure and that the complexity of it is $O(k \cdot n)$, where k is the number of sub-formulae and n the maximal number of positive intervals in the atomic signals.

2.3 REAL-VALUED SIGNALS

Extend our semantics and logic to real-valued signals. Unlike Boolean and finite variability signals, the real-valued ones do not admit a finite representation. They are represented via sampling that is a sequence of time stamped values of the form $(t, s[t])$.
For signals of the form $s : \mathbb{T} \to \mathbb{R}^m$ is not possible to ignore issues related to their representation based on output of some numeric simulator. The logic do not speak about continuous signals but rather of a set of static abstraction from $\mu : \mathbb{R}^m \to \mathbb{B}$.

This function create a partition of the continuous state-space according to the satisfaction of some inequality constraints on the real variables. As long as $\mu(s[t])$ is constant we do not care about the exact value of the signal. However, in order to be able to sample the formulae we need to require our sampling to be sufficiently dense, in order to detect the transition.
The dynamics of most reasonable systems have bounded frequency, and even adding noise, the frequency remains bounded by the size of the integration step by the simulator. We assume we deal with signals that are well-behaving with respect to every μ, which means that:

- $\mu(s)$ has a bounded variability,
- every change in $\mu(s)$ is detected, i.e. every point t s.t. $\mu(s[t]) \neq \lim_{t' \to t} \mu(s[t'])$ is included in the sampling.

Definition. *Let $U = \{\mu_1, \ldots, \mu_n\}$ be a collection of predicates (effective functions) of the form $\mu_i : \mathbb{R}^m \to \mathbb{B}$. Then an $\mathtt{STL}(U)$ formula is an $\mathtt{MITL}_{[a,b]}$ formula over the atomic propositions $\mu_1(x), \ldots, \mu_n(x)$.*

Any well-behaved signal with respect to U can be transformed into a Boolean signal $s' : \mathbb{T} \to \mathbb{B}^n$ s.t. $s' = \mu_1(s) \| \mu_2(s) \| \ldots \| \mu_{n-1}(s) \| \mu_n(s)$ is of bounded variability. By construction, we get that for every STL formula φ, $s \vDash \varphi$ iff $s' \vDash \varphi'$ in $\mathtt{MITL}_{[a,b]}$. (φ' is obtained by φ by replacing every $\mu_i(x)$ with a propositional variable p_i)
The monitoring process is the following:

1. Construction of a Boolean filter for every $\mu_i \in U$,
2. Transform s into a Boolean signal, i.e. $p_i = \mu_i(s)$

Ex. Consider the signal $\sin[t]$ where t is measured in degrees and $\mu(x) = x > 0$. The sampling is of length 400, collecting measure every 50 time units, plus the collections of the zeros of the function at 180 and 360 The input of the Boolean filter is the following:

$$(0, 0.0), (50, 0.766), (100, 0.984), (150, 0.5), (180, 0.0), (200, -0.342),$$
$$(250, -0.939), (300, -0.866), (350, -0.173), (360, 0), (400, 0.643)$$

and the output is the signal p s.t. $\mathcal{I}_p^+ = \{[0, 180), [360, 400)\}$, then we can apply the monitoring described before.

3 "Efficient robust monitoring for STL"

Object: Presentation of an efficient algorithm for computing the robustness degree in which piecewise-continuous signal satisfies or violates a STL formula.

TL is a popular formalism used in systems design and formal verification in an either deductive or algorithmic method. The behaviors studied are typically discrete (sequence of states or events). Problem with the assumption behind TL use:

- **Correctness**: requires that all system behaviors satisfy the specification, thus model checking is based on composing the system model with an automaton for specification and analyzing and all possible paths
- **Discreteness**: the satisfaction in this logic is purely discrete.

New trends in alternative ways of using TL in designing complex systems, including their operations:

1. **Statistical methods a-la Monte Carlo** where universal quantification is replaces with random simulation.
2. **Assertion checking or monitoring** where TL is used as a rigorous specification of the requirements, but it is evaluated on a single behavior at a time. It requires a process that generates observable behaviors. Temporal property checking can be integrated in monitoring and diagnostics of real systems during their operation

Monitoring: does not require any model, only a process that generates observable behavior. It can be applied to systems as black boxes or to programs without a decent and tractable formal model. The temporal propriety monitoring can be implemented in real systems.

In the paper is presented STL and its monitoring AMT. When dealing with continuous dynamics and numerical quantities, yes/no answers provide only partial information and could be augmented with quantitative information about the satisfaction to provide a better basis for decision-making.
Ex. Consider $\varphi : \mathsf{x} < c$ with c constant and $x \in X \subseteq \mathbb{R}$. The formula φ we can define a cut on the set in $X_0 = \{x \mid x \geq c\}$ and $X_1 = \{x \mid x < c\}$. X_1 is called the **validity domain** of the formula, since we have that if there is an $x \in X$ s.t. $x \vDash \mathsf{x} < c$, then it means that $x \in X_1$. The robustness degree of a satisfaction should tell us if x satisfies the formula by far or marginally. It captures the $c - x$ and its sign tell us the satisfaction violation of the formula, while its magnitude the robustness.
In this paper an optimal algorithm that computes robustness is presented. This algorithm guarantees that the over-head added by monitoring to the simulation process is acceptable. It computes in linear time, fo the following reasons:

- use of the optimal streaming algorithm of Daniel Lemire to compute the extrema of a sequence over an interval,
- rewriting of the bounded until operator as a conjunction of an untimed and a simple timed operators.

3.1 Signal Temporal Logic

Recall the framework set of [MN04] and extension of it to a multivalued logic as proposed in [FP09]. The set of Booleans is defined as follows $\mathbb{B} = \{\bot, \top\}$, with the following proprieties:

- $\bot < \top$,
- $-\top = \bot$,
- $-\bot = \top$.

Consider $\overline{\mathbb{R}} = \mathbb{R} \cup \mathbb{B}$ that is a total-ordered set of real numbers with the smallest element $\bot$ and greatest $\top$.
A signal is a function $s : \mathcal{D} \to \mathcal{E}$, with $\mathcal{D} \subseteq \mathbb{R}_0^+$ and $\mathcal{E} \subset \overline{\mathbb{R}}$. If $\mathcal{E} = \mathbb{B}$ then s is a Boolean signal, if $\mathcal{E} = \mathbb{R}$ is a real-valued signal.
A trace w is a set of real-valued signals $\{x_1^w, \ldots, x_k^w\}$ defined some interval $\mathcal{D}$, called time domain of w. We can transform such trace in a set of predicates of the form $x_i \geq 0$. STL is an extension of Metric Temporal Logic using real-valued signals.

$$\textbf{Syntax:} \quad \varphi := \top \mid x_i \geq 0 \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \mathcal{U}_{[a,b]} \varphi_2$$

Let w be a trace of time domain $\mathcal{D}$. The formula φ is said defined over a time interval $\mathrm{dom}(\varphi, w)$, define recursively on the complexity of φ as follows:

$$\begin{aligned}
\mathrm{dom}(\top, w) &= \mathrm{dom}(x_i \geq 0, w) = \mathcal{D}, \\
\mathrm{dom}(\neg\varphi, w) &= \mathrm{dom}(\varphi, w), \\
\mathrm{dom}(\varphi \wedge \psi, w) &= \mathrm{dom}(\varphi, w) \cap \mathrm{dom}(\psi, w), \\
\mathrm{dom}(\varphi \vee \psi, w) &= \mathrm{dom}(\varphi, w) \cup \mathrm{dom}(\psi, w), \\
\mathrm{dom}(\varphi \,\mathcal{U}_{[a,b]}\, \psi, w) &= \{t \in \mathbb{R} \mid t, t + \inf[a, b] \in \mathrm{dom}(\varphi, w) \text{ and } \\
&\qquad t + \inf[a, b] \in \mathrm{dom}(\psi, w)\}.
\end{aligned}$$

3.1.1 Boolean semantics

Now, we can define the Boolean semantics. Given a trace w, the validity of a STL formula φ at a time $t \in \mathrm{dom}(\varphi, w)$, is defined inductively as follows:

$$\begin{aligned}
w, t &\vDash \text{true} \\
w, t \vDash x_i \geq 0 &\quad \Leftrightarrow \quad x_i^w(t) \geq 0 \\
w, t \vDash \neg\varphi &\quad \Leftrightarrow \quad w, t \nvDash \varphi \\
w, t \vDash \varphi \wedge \psi &\quad \Leftrightarrow \quad w, t \vDash \varphi \text{ and } w, t \vDash \psi \\
w, t \vDash \varphi \vee \psi &\quad \Leftrightarrow \quad w, t \vDash \varphi \text{ or } w, t \vDash \psi \\
w, t \vDash \varphi \,\mathcal{U}_{[a,b]}\, \psi &\quad \Leftrightarrow \quad \exists t' \in [a + t, b + t], w, t' \vDash \psi \text{ and } \forall t'' \in [t, t'] w, t'' \vDash \varphi_1[t'']
\end{aligned}$$

All the abbreviations are valid as well as the definitions for $\Box_{[a,b]}\, \varphi$ and $\Diamond_{[a,b]}\, \varphi$.
We have also the untimed operator that are timed operators over the interval

$[0, \infty)$. Then we give also the following definition:

$$\text{false} = \neg\text{true}.$$

For a given formula φ and a trace w, we define the **satisfaction signal** $\chi(\varphi, w, \bullet)$ as follows:

$$\text{for all } t \in \text{dom}(\varphi, w), \quad \chi(\varphi, w, t) = \begin{cases} \top & \text{if } w, t \vDash \varphi, \\ \bot & \text{otherwise.} \end{cases}$$

Monitoring the satisfaction of a formula φ can be done by computing for each subformula the entire satisfaction signal $\chi(\varphi, w, \bullet)$, following the procedure described in [MN04].

3.1.2 Quantitative semantics

Given a formula φ, a trace w and a time $t \in \text{dom}(\varphi, w)$, we define the quantitative semantics $\rho(\varphi, w, t)$ inductively as follows:

$$\begin{aligned}
\rho(\text{true}, w, t) &= \top \\
\rho(x_i \geq 0, w, t) &= x_i^w(t) \\
\rho(\neg\varphi, w, t) &= -\rho(\varphi, w, t) \\
\rho(\varphi \wedge \psi, w, t) &= \min\{\rho(\varphi, w, t), \rho(\psi, w, t)\} \\
\rho(\varphi \vee \psi, w, t) &= \max\{\rho(\varphi, w, t), \rho(\psi, w, t)\} \\
\rho(\varphi \,\mathcal{U}_{[a,b]}\, \psi) &= \sup_{t' \in [a+t, b+t]} \min\left\{\rho(\psi, w, t'), \inf_{t'' \in [t, t']} \rho(\varphi, w, t'')\right\}
\end{aligned}$$

If we define $\chi(x_i > 0, w, t)$, applying the definition above and we apply it to the inductive definition of ρ we get back to Boolean signals and obtain again χ.
In this type of semantics however, atomic predicates do not evaluate $\top$ or $\bot$, but give a real value representing the distance to satisfaction or to violation and propagate it to the formula using operations on $\overline{\mathbb{R}}$.
For the lattice proprieties of $(\overline{\mathbb{R}}, <)$ we are granted the associativity, commutativity, neutral element and distributivity. Minus function is involutive, which gives the propositional and the modal version of de Morgan laws:

$$\begin{aligned}
\neg(\varphi \vee \psi) &\equiv \neg\varphi \wedge \neg\psi \\
\neg \Diamond_{[a,b]} \varphi &\equiv \Box_{[a,b]} \neg\varphi
\end{aligned}$$

Therefore, we have for the derived operators the same natural interpretations:

$$\begin{aligned}
\rho(\Diamond_{[a,b]} \varphi, w, t) &= \sup_{t' \in [a+t, b+t]} \rho(\varphi, w, t') \\
\rho(\Box_{[a,b]} \varphi, w, t) &= \inf_{t' \in [a+t, b+t]} \rho(\varphi, w, t')
\end{aligned}$$

3.1.3 Propriety of Robustness Estimate

The quantitative semantics of STL have two proprieties, that would alone justify their introduction.
If $\rho(\varphi, w, t) \neq 0$, its sign indicates the satisfaction status.

Proposition (Soundness)**.** *Let φ be a* STL *formula, w a trace and a t time. Then*

$$\rho(\varphi, w, t) > 0 \quad \Rightarrow \quad w, t \vDash \varphi$$

If w satisfies φ at time t, any other trace w' whose pointwise distance from w is smaller than $\rho(\varphi), w, t$ also satisfies φ at time t.

Proposition. *(Correctness) Let φ be an* STL *formula, w and w' traces over the same time domain, and $t \in \mathrm{dom}(\varphi, w)$. Then*

$$w, t \vDash \varphi \text{ and } \|w - w'\|_\infty < \rho(\varphi, w, t) \quad \Rightarrow \quad w', t \vDash \varphi$$

Given the soundness and the correctness, for a given trace w, and a STL formula φ, we have that a **robustness signal** of φ w.r.t. w, $\rho(\varphi, w, \bullet)$, is defined over the domain $\mathrm{dom}(\varphi, w)$.

3.1.4 Rewriting the operator Until

We can claim about the until operator the following proprieties, extended from the Boolean to the quantitative semantics:

$$\varphi \,\mathcal{U}_{[a,b]}\, \psi \quad \equiv \quad \Diamond_{[a,b]}\, \psi \,\wedge\, \varphi \,\mathcal{U}_{[a,+\infty)}\, \psi \tag{3.1}$$

$$\varphi \,\mathcal{U}_{[a,+\infty)}\, \psi \quad \equiv \quad \Box_{[0,a]} (\varphi \,\mathcal{U}\, \psi) \tag{3.2}$$

Proof.
(3.1)
Let be y, y' the robustness signals φ, ψ relative to w. Now consider

$$u = \sup_{\tau \in [a+t, b+t]} \min\left\{ y'(\tau), \inf_{[t,\tau]} y \right\} \quad \text{and}$$

$$v = \min\left\{ \sup_{[a+t,b+t]} y', \sup_{\tau \geq t+a} \min\left\{ y'(\tau), \inf_{[t.\tau]} y \right\} \right\}$$

and let then be the robustness values of the first equivalence for some given t. Suppose $u \neq v$. Then we define the signals

$$x \,:\, t \mapsto y(t) - \frac{u+v}{2} \quad \text{and} \quad x' \,:\, t \mapsto y'(t) - \frac{u+v}{2}$$

Let be the formualae

$$\gamma = (x \geq 0)\, \mathcal{U}_{[a,b]} (x' \geq 0) \quad \text{and} \quad \vartheta = \Diamond_{[a,b]} (x \geq 0) \wedge (x \geq 0)\, \mathcal{U}_{[a,+\infty)} (x' \geq 0)$$

□

It is possible to push the constant outside the scope of the operators of min, sup, inf. Doing this we get their quantitative semantics:

$$\rho(\gamma, w, t) = v - \frac{u+v}{2} < 0 \qquad \text{and} \quad \rho(\vartheta, w, t) = v - \frac{u+v}{2} > 0$$

$$\xRightarrow{\text{By Soundness}} \quad w, t \nvDash \gamma \qquad\qquad \text{and} \quad w, t \vDash \vartheta$$

3.2 Computing the Robustness Estimate

In monitoring signals are available as finite timed over an alphabet $\mathbb{R}^n$, and they can be interpreted as linear interpolation. Presentation of the basic framework for computing robustness, under this hypothesis.

Algorithm 1: Robustness(φ, w)

```
switch φ do
    case true do
        return ⊤̄   % a constant ⊤̄ signal;
    end
    case x_i ≥ 0 do
        return x_i^w;
    end
    case ⊙ φ_1 do
        y ← Robustness(φ_1, w);
        return Compute(⊙, y);
    end
    case φ_1 ⊙ φ_2 do
        y ← Robustness(φ_1, w);
        y' ← Robustness(φ_2, w);
        return Compute(⊙, y, y');
    end
end
```

Definition. *A signal y is said to be **finitely piecewise-linear, continuous*** (f.p.l.c.) *if there exists a finite sequence $(t_i)_{i\leq n_y}$ s.t.:*

- *y is defined over the domain $[t_0, t_{n_y})$,*
- *for all $i < n_y$, y is continuous at t_i and offline on $[t_i, t_{i+1})$*

*$(t_i)_{i\leq n_y}$ is called the **time sequence** of y.*

Let be $\partial y(t) = \frac{\partial}{\partial t} y(t)$. Any signal in the observed trace will be assumed to be f.p.l.c., and will be represented by the sequence $(t_i, y(t_i), \partial y(t_i))_{i<n_y}$, along with the cut-off time t_{n_y}.
The representation is redundant, by continuity, but facilitates the splitting into segments. The quantitative semantics preserves the f.p.l.c. proprieties of the signal. Continuity is preserved over the operators sup and inf, and no new derivative value is created in the process.

3.2.1 Boolean operators

Computing the robustness of $\neg\varphi$ from φ is trivial. Note that if the sequence $(t_i, y(t_i), \partial y(t_i))_{i<n_y}$ represents $\rho(\varphi, w, \bullet)$ then the sequence $(t_i, -y(t_i), -\partial y(t_i))_{i<n_y}$ represents $\rho(\neg\varphi, w, \bullet)$.
For conjunction let take y and y' as the robustness of φ and ψ respectively, producing z the robustness signal of $\varphi \wedge \psi$. Then consider the sequence $(r_i)_{i\leq n_z}$ containing the sampling points of y and y' when they are both defined, and the points where y and y' intersect.

Bibliography

[DFM13] Alexandre Donzé, Thomas Ferrere, and Oded Maler. "Efficient robust monitoring for STL". In: *Computer Aided Verification: 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings 25*. Springer. 2013, pp. 264–279.

[DM10] Alexandre Donzé and Oded Maler. "Robust satisfaction of temporal logic over real-valued signals". In: *International Conference on Formal Modeling and Analysis of Timed Systems*. Springer. 2010, pp. 92–106.

[Don+12] Alexandre Donzé et al. "On temporal logic and signal processing". In: *Automated Technology for Verification and Analysis: 10th International Symposium, ATVA 2012, Thiruvananthapuram, India, October 3-6, 2012. Proceedings 10*. Springer. 2012, pp. 92–106.

[FP09] Georgios E Fainekos and George J Pappas. "Robustness of temporal logic specifications for continuous-time signals". In: *Theoretical Computer Science* 410.42 (2009), pp. 4262–4291.

[MN04] Oded Maler and Dejan Nickovic. "Monitoring temporal properties of continuous signals". In: *International symposium on formal techniques in real-time and fault-tolerant systems*. Springer. 2004, pp. 152–166.

[MNP08] Oded Maler, Dejan Nickovic, and Amir Pnueli. "Checking temporal properties of discrete, timed and continuous behaviors". In: *Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday* (2008), pp. 475–505.

[NM07] Dejan Nickovic and Oded Maler. "AMT: A property-based monitoring tool for analog systems". In: *International Conference on Formal Modeling and Analysis of Timed Systems*. Springer. 2007, pp. 304–319.