A General Framework for Robust Quantitative Semantics of Signal Temporal Logic

Jiawei Chen[1][0000−0002−5461−6711],
José Luiz Vargas de Mendonça[1][0000−0002−1576−2492],
Konstantinos Mamouras[2][0000−0003−1209−7738], and
Jean-Baptiste Jeannin[1][0000−0001−6378−1447]

[1] University of Michigan
[2] Rice University

Abstract. Quantitative semantics of Signal Temporal Logic (STL) play an important role in both the falsification and control synthesis for dynamical systems by assigning numerical quantities to truth values. Recently, several different quantitative semantics have been proposed, offering better performance in many cases. Yet a general, systematic understanding of the structure and properties of quantitative semantics is missing. In this paper, we develop a general framework to model quantitative semantics. We focus mainly on soundness, which requires that the quantitative semantics of a statement is positive when the statement is true, and negative when the statement is false. This ensures that counterexamples will not be missed during verification. We derive simple, necessary conditions in our framework for soundness. We show how several recently proposed quantitative semantics fit in our framework, and how others do not, typically because they do not strictly satisfy soundness. We implement various quantitative semantics, including existing semantics from literature, in our framework and compare their effectiveness as objective functions for optimization-based falsification on both novel and existing benchmarks.

Keywords: Signal Temporal Logic · Quantitative Semantics · Cyber-Physical Systems

1 Introduction

Metric temporal logic (MTL) [23] and signal temporal logic (STL) [27] are widely used for specifying correctness properties for cyber-physical systems [6]. The standard semantics for MTL and STL is qualitative, specifically Boolean. The Boolean semantics only tells us whether a property φ holds or not with respect to a system trace. This is, however, not informative enough. In many cases, we would also like to know how *robustly* the system trace satisfies (or falsifies) a desired property. This can be useful, for example, when determining the degree to which a system's safety is resilient against unexpected disturbances. Faella, Legay, and Stoelinga [14] propose a quantitative framework for linear temporal

logic (LTL) that can be verified via model checking. Fainekos and Pappas [15] define a real-valued quantitative semantics for MTL and STL that captures the notion of robustness of satisfaction.

In the semantics of Fainekos and Pappas, a truth value is a real number or $\pm\infty$ with the following interpretation: a positive number indicates truth, a negative number indicates falsity, and the truth value 0 is ambiguous. The logical and temporal connectives are interpreted using the operation min for conjunction and max for disjunction. Negation is interpreted using the unary $-$ operation. A key property of this min-max semantics is that it is *sound*, which roughly means that it agrees with the standard qualitative (i.e., Boolean) semantics. If ρ is the min-max semantics of Fainekos and Pappas and the Boolean semantics is given with the satisfaction relation (denoted by $\models$), then the soundness property can be formulated as follows:

1. $\rho(\varphi, \sigma, t) > 0$ implies $\sigma^t \models \varphi$ and
2. $\rho(\varphi, \sigma, t) < 0$ implies $\sigma^t \not\models \varphi$

for every temporal formula φ, trace σ and time instant t, where we write $\sigma^t \models \varphi$ to mean that the trace σ satisfies the formula φ at time instant t.

The real-valued quantitative semantics of MTL/STL has found use in the *falsification* of cyber-physical systems, which is about finding design errors using formal specifications of correct/safe system behavior. Given a system model M and a temporal formula φ, the falsification problem asks whether there exists an input signal σ_i so that the system trace $M(\sigma_i)$ falsifies the desired property φ. Solving this problem amounts to a search in the space of input signals. When a (sound) quantitative semantics is available, then falsification can be reduced to a minimization problem: find an input signal σ_i that minimizes the truth value $\rho(\varphi, M(\sigma_i))$. Several different optimization approaches have been used to solve this problem: Monte-Carlo stochastic optimization [32,1], cross-entropy [34], and other iterative approaches [2]. Breach [11] and S-TaLiRo [4] are two state-of-the-art tools for the falsification of temporal properties of cyber-physical systems.

The robustness (quantitative) semantics of MTL and STL are also useful for solving the controller synthesis problem for cyber-physical systems, where the goal is to find parameter values for a controller so that the system satisfies the specification as robustly as possible [33]. The controller synthesis problem can be cast as a robustness maximization problem.

After the proposed min-max quantitative semantics of Fainekos and Pappas [15], which is often described as "spatial" robustness, several other quantitative semantics have been proposed for MTL and STL. Donzé and Maler [10] consider an extension of spatial robustness that also accounts for temporal displacement. Akazaki and Hasuo [3] proposed an extension of MITL with averaged temporal operators. Another average-based robustness was explored in [30,26]. The original robustness semantics [15] uses max (resp., min) for interpreting disjunction (resp., conjunction), which are not smooth functions. Since smoothness is a valuable property in the context of falsification and synthesis, many authors have considered smooth variants of the robustness semantics [33,17,16].

Similarly, the TeLEx tool [21] uses a smooth version of a *tightness* semantics in order to learn STL formulas from only positive examples. TeLEx uses smooth functions for the temporal operators and atomic predicates, which means that the semantics is differentiable. This smoothness property enables the effective use of gradient-based optimization methods.

Recent developments [35,36] employ Monte Carlo Tree Search to overcome some challenges of a purely optimization-based approach to falsification, in particular the "scale problem" when combining formulas. Nevertheless, these methods still employ quantitative semantics, so the generalization of these semantics, and especially the formalization of soundness conditions, remains relevant.

This proliferation of quantitative semantics for MTL/STL raises the question of whether it is possible to develop a general mathematical framework that (1) can describe a large class of quantitative semantics, and (2) encompasses previously considered semantics. In the context of such a framework, it would be important to develop a methodology for easily checking different properties of a semantics, especially its soundness. Soundness is a fundamental property that ensures agreement with the standard Boolean semantics and is essential for avoiding incorrect results, especially in the context of falsification. A generalization of quantitative semantics has been proposed previously [19,5], based on semirings and an automata-based construction, but it is limited to discrete-time models. Monitoring with algebraic robustness semantics is considered in [8,29] (using lattices) and in [28] (using semirings). In contrast, our proposed generalization and soundness criteria can be extended to continuous-time temporal logics, and operate directly on the original unmodified specification. While many of the semantics we examined may be implemented in the aforementioned discrete-time framework, there exist some which benefit from a more granular approach that allows mixing of different integrators of the same subclass, e.g. using a combination of min and multiplication. In addition, we introduce soundness conditions to check the individual member functions in our generalization. Regardless of the method, a general framework is important for methodical comparisons of semantics, independent of their original implementations.

Contributions Our main contributions extend previous work-in-progress [20] in the following ways:

- We describe a mathematical framework for robustness quantitative semantics for MTL/STL that allows us to easily check that a semantics is sound.
- We show how this new framework can generalize many generalizations of quantitative semantics that are found in previous work.
- Using our framework, we identify a mistake in previous work [17], showing the effectiveness of our framework at ensuring soundness of quantitative semantics. We also show how to fix that semantics.
- Using Breach, we compare experimentally the effectiveness of several different robustness semantics in the context of CPS falsification.

Paper outline Section 2 gives some preliminary definitions for the syntax of MTL/STL, its standard qualitative (Boolean) semantics, and the widely used

min-max quantitative semantics of [15]. Section 3 presents our generalization of robustness semantics for MTL/STL and Section 4 details sufficient conditions for a semantics to be sound. In Section 5, we present an experimental comparison of robustness semantics in the context of CPS falsification.

2 A Generalization of Quantitative Semantics

2.1 Syntax and Qualitative Semantics

We consider properties of a real-valued signal over a discrete or continuous domain $\mathbb{T}$. In the continuous-time case, $\mathbb{T} = \mathbb{R}_+$; in the discrete-time case $\mathbb{T} = \mathbb{N}$. We denote by I a bounded interval over $\mathbb{T}$, and by $\mathbb{I}$ the set of all such bounded intervals. In the continuous case, we assume that the signal is integrable on any bounded interval $I \in \mathbb{I}$. A signal σ over n variables $x_1, ..., x_n$ is a function from $\mathbb{T}$ to $\mathbb{R}^n$. We write $l(\sigma)$ to mean a function of those n variables with real values. We also write $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ and similarly $\overline{\mathbb{R}}_+ = \mathbb{R}_+ \cup \{+\infty\}$ and $\overline{\mathbb{R}}_- = \mathbb{R}_- \cup \{-\infty\}$.

Syntax. An STL formula φ is given by:

$$\varphi, \psi ::= \top \mid l(\sigma) \geq 0 \mid l(\sigma) > 0 \mid \neg\varphi \mid \varphi \wedge \psi \mid \psi \mathbf{U}_I \varphi$$

The qualitative semantics of STL, $\sigma^t \vDash \varphi$, defines the truth value of σ at time t with respect to φ, and is standard [15,10]:

- It is always true that $\sigma^t \vDash \top$.
- $\sigma^t \vDash l(\sigma) \geq 0$ (resp. > 0) if and only if the value of l at time t, $l(\sigma[t]) \geq 0$ (resp. > 0).
- $\sigma^t \vDash \neg\varphi$ when $\sigma^t \nvDash \varphi$.
- $\sigma^t \vDash \varphi_1 \wedge \varphi_2$ when $\sigma^t \vDash \varphi_1$ and $\sigma^t \vDash \varphi_2$.
- $\sigma^t \vDash \varphi_1 \mathbf{U}_I \varphi_2$ if and only if there exists $t' \in I$ such that $\sigma^{t+t'} \vDash \varphi_2$ and for all $t'' \in [t, t + t']$, $\sigma^{t''} \vDash \varphi_1$.

Operators $\bot$, $\vee$, $\mathbf{F}_I$ and $\mathbf{G}_I$ are derived from the above using the standard equivalences.

2.2 Standard Quantitative Semantics

The standard quantitative semantics $\rho_0(\varphi, \sigma, t) \in \overline{\mathbb{R}}$ is defined for a specification φ, a trace σ over $\mathbb{T}$, and a time $t \in \mathbb{T}$ [15,10]. It is convenient to split ρ_0 into its positive and negative parts $\rho_0^+ \in \overline{\mathbb{R}}_+$ and $\rho_0^- \in \overline{\mathbb{R}}_-$ to facilitate adding robustness values, following the convention of existing literature [3,17]. Here, $\rho_0(\varphi, \sigma, t) = \rho_0^+(\varphi, \sigma, t) + \rho_0^-(\varphi, \sigma, t)$, where ρ_0^+ and ρ_0^- are:

$$
\begin{aligned}
\rho_0^+(\top,\sigma,t) &= +\infty \\
\rho_0^-(\top,\sigma,t) &= 0 \\
\rho_0^+(l(\sigma)\geq 0,\sigma,t) &= \rho_0^+(l(\sigma)>0,\sigma,t) = \max(0,l(\sigma[t])) \\
\rho_0^-(l(\sigma)\geq 0,\sigma,t) &= \rho_0^-(l(\sigma)>0,\sigma,t) = \min(0,l(\sigma[t])) \\
\rho_0^+(\neg\varphi,\sigma,t) &= -\rho_0^-(\varphi,\sigma,t) \\
\rho_0^-(\neg\varphi,\sigma,t) &= -\rho_0^+(\varphi,\sigma,t) \\
\rho_0^+(\varphi\wedge\psi,\sigma,t) &= \min(\rho_0^+(\varphi,\sigma,t),\rho_0^+(\psi,\sigma,t)) \\
\rho_0^-(\varphi\wedge\psi,\sigma,t) &= \min(\rho_0^-(\varphi,\sigma,t),\rho_0^-(\psi,\sigma,t)) \\
\rho_0^+(\psi\mathbf{U}_I\varphi,\sigma,t) &= \sup_{t'\in I}\left(\min(\rho_0^+(\varphi,\sigma,t+t'),\inf_{t''\in[t,t+t']}\rho_0^+(\psi,\sigma,t''))\right) \\
\rho_0^-(\psi\mathbf{U}_I\varphi,\sigma,t) &= \sup_{t'\in I}\left(\min(\rho_0^-(\varphi,\sigma,t+t'),\inf_{t''\in[t,t+t']}\rho_0^-(\psi,\sigma,t''))\right)
\end{aligned}
$$

An essential property of quantitative semantics is soundness:

Theorem 1. *if* $\rho_0(\varphi,\sigma,t)>0$ *then* $\sigma^t\vDash\varphi$*, and if* $\rho_0(\varphi,\sigma,t)<0$ *then* $\sigma^t\nvDash\varphi$.

Proof. This result is proved by structural induction on the STL formula [15,10].

2.3 A general framework

A goal of this work is to establish a generalized form for quantitative semantics, parameterized by unary functions $\nu:\mathbb{R}\to\overline{\mathbb{R}}_+$ and $\mu:\mathbb{R}\to\overline{\mathbb{R}}_-$, binary integrators α, β, ζ, $\eta:\overline{\mathbb{R}}_+\times\overline{\mathbb{R}}_+\to\overline{\mathbb{R}}_+$, as well as time integrators Γ, Δ, Θ, $\Xi:\mathbb{I}\times(\mathbb{T}\to\overline{\mathbb{R}}_+)\to\overline{\mathbb{R}}_+$. Using those operators, we can define a generic $\rho(\varphi,\sigma,t)=\rho^+(\varphi,\sigma,t)+\rho^-(\varphi,\sigma,t)$ with:

$$
\begin{aligned}
\rho^+(\top,\sigma,t) &= +\infty \\
\rho^-(\top,\sigma,t) &= 0 \\
\rho^+(l(\sigma)\geq 0,\sigma,t) &= \rho^+(l(\sigma)>0,\sigma,t) = \nu(l(\sigma[t])) \\
\rho^-(l(\sigma)\geq 0,\sigma,t) &= \rho^-(l(\sigma)>0,\sigma,t) = \mu(l(\sigma[t])) \\
\rho^+(\neg\varphi,\sigma,t) &= -\rho^-(\varphi,\sigma,t) \\
\rho^-(\neg\varphi,\sigma,t) &= -\rho^+(\varphi,\sigma,t) \\
\rho^+(\varphi\wedge\psi,\sigma,t) &= \alpha(\rho^+(\varphi,\sigma,t),\rho^+(\psi,\sigma,t)) \\
\rho^-(\varphi\wedge\psi,\sigma,t) &= -\beta(-\rho^-(\varphi,\sigma,t),-\rho^-(\psi,\sigma,t)) \\
\rho^+(\psi\mathbf{U}_I\varphi,\sigma,t) &= \prod_{t'\in I}\zeta\left(\rho^+(\varphi,\sigma,t+t'),\bigtriangleup_{t''\in[t,t+t']}\rho^+(\psi,\sigma,t'')\right) \\
\rho^-(\psi\mathbf{U}_I\varphi,\sigma,t) &= -\bigoplus_{t'\in I}\eta\left(-\rho^-(\varphi,\sigma,t+t'),\Xi_{t''\in[t,t+t']}-\rho^-(\psi,\sigma,t'')\right)
\end{aligned}
$$

Note that, as also mentioned in related work, the quantitative semantics of the formulas $f(\sigma) \geq 0$ and $f(\sigma) > 0$ are the same, hence those two formulas cannot be distinguished from the value of their quantitative semantics alone [3].

In order to clarify the presentation, we follow Haghighi et al. [17] in writing $\rho(\varphi, \sigma, t) = \rho^+(\varphi, \sigma, t) + \rho^-(\varphi, \sigma, t)$, and we have the following properties, which we can prove by induction on the definitions of ρ, ρ^+ and ρ^-:

Lemma 1. *For any φ, σ and t, $\rho^+(\varphi, \sigma, t) \geq 0$ and $\rho^-(\varphi, \sigma, t) \leq 0$.*

Proof. By structural induction and using the sign constraints on the codomains of ν, μ, α, β, ζ, η, $\prod$, Δ, Θ and Ξ. A detailed proof is available in the extended version of the paper.

From those definitions we can derive the definition of $\bot$ and $\vee$ from $\wedge$ and $\neg$; and $\mathbf{F}_I$ and $\mathbf{G}_I$ from $\mathbf{U}_I$, under mild assumptions on ζ, η, Δ and Ξ:

$$\begin{aligned}
\rho^+(\bot, \sigma, t_k) &= \rho^+(\neg\top, \sigma, t_k) = 0 \\
\rho^-(\bot, \sigma, t_k) &= \rho^-(\neg\top, \sigma, t_k) = -\infty \\
\rho^+(\varphi \vee \psi, \sigma, t_k) &= \rho^+(\neg(\neg\varphi \wedge \neg\psi), \sigma, t_k) = \beta(\rho^+(\varphi, \sigma, t_k), \rho^+(\psi, \sigma, t_k)) \\
\rho^-(\varphi \vee \psi, \sigma, t_k) &= \rho^-(\neg(\neg\varphi \wedge \neg\psi), \sigma, t_k) = -\alpha(-\rho^-(\varphi, \sigma, t_k), -\rho^-(\psi, \sigma, t_k)) \\
\rho^+(\mathbf{F}_I\varphi, \sigma, t_k) &= \rho^+(\top\mathbf{U}_I\varphi, \sigma, t_k) = \prod_{k' \in I} \zeta\left(\rho^+(\varphi, \sigma, t_{k+k'}), \Delta_{k'' \in [k, k+k']} + \infty\right) \\
\rho^-(\mathbf{F}_I\varphi, \sigma, t_k) &= \rho^-(\top\mathbf{U}_I\varphi, \sigma, t_k) = -\Theta_{k' \in I}\, \eta\left(-\rho^-(\varphi, \sigma, t_{k+k'}), \Xi_{k'' \in [k, k+k']} 0\right) \\
\rho^+(\mathbf{G}_I\varphi, \sigma, t_k) &= \rho^+(\neg\mathbf{F}_I\neg\varphi, \sigma, t_k) = \Theta_{k' \in I}\, \eta\left(\rho^+(\varphi, \sigma, t_{k+k'}), \Xi_{k'' \in [k, k+k']} 0\right) \\
\rho^-(\mathbf{G}_I\varphi, \sigma, t_k) &= \rho^-(\neg\mathbf{F}_I\neg\varphi, \sigma, t_k) = -\prod_{k' \in I} \zeta\left(-\rho^-(\varphi, \sigma, t_{k+k'}), \Delta_{k'' \in [k, k+k']} + \infty\right)
\end{aligned}$$

Remark 1. As long as $\Delta_{k'' \in [k, k+k']} + \infty = +\infty$ and $\zeta(x, +\infty) = x$ for any $x \geq 0$, we can simplify the following two definitions:

$$\begin{aligned}
\rho^+(\mathbf{F}_I\varphi, \sigma, t_k) &= \prod_{k' \in I} \rho^+(\varphi, \sigma, t_{k+k'}) \\
\rho^-(\mathbf{G}_I\varphi, \sigma, t_k) &= -\prod_{k' \in I} -\rho^-(\varphi, \sigma, t_{k+k'})
\end{aligned}$$

Remark 2. As long as $\Xi_{k'' \in [k, k+k']} 0 = 0$ and $\eta(x, 0) = x$ for any $x \geq 0$, we can simplify the following two definitions:

$$\begin{aligned}
\rho^-(\mathbf{F}_I\varphi, \sigma, t_k) &= -\Theta_{k' \in I} -\rho^-(\varphi, \sigma, t_{k+k'}) \\
\rho^+(\mathbf{G}_I\varphi, \sigma, t_k) &= \Theta_{k' \in I} \rho^+(\varphi, \sigma, t_{k+k'})
\end{aligned}$$

3 Semantics Instantiations

Our generalization can instantiate semantics that replicate or closely match those presented in existing literature, in terms of the aforementioned generic functions (ν, μ, α, β, ζ, η, Γ, Δ, Θ, Ξ). The particular instantiations we chose to represent each semantics is shown in Table 1.

Several semantics were already implemented in Breach, which we adapted to our generalization by splitting the computation of ρ into that of ρ^+ and ρ^-. **Max** [11] emulates the default Breach semantics, which utilizes `min` and `max` for integrators, and `min` and `max` with respect to 0 for the "rectifiers" ν and μ which help split the single robustness scores into positive and negative components.

Const [11] uses the same integrators as Max, but the rectifier functions ν and μ produce constants which depend only on the sign of the predicate's robustness score. This is represented by $A \times \text{sgn}(x)$, where A is a fixed constant ($A = 100$ in the original Breach implementation, which we replicate), and $\text{sgn}(x)$ returns 1 if $x > 0$, 0 if $x = 0$, and -1 if $x < 0$. This simulates Boolean STL semantics, as all scores have the same absolute value regardless of their degree of satisfaction[11].

Add [11] is similar to Max except in certain cases where `max` is replaced by the "Koen &+" operator [9,25]. This integrator is a piecewise function $f(x, y)$ that takes on the behavior of the harmonic mean in certain cases, addition in others, and min everywhere else:

$$f(x,y) = \begin{cases} x + y & x < 0 \wedge y < 0 \\ \frac{1}{\frac{1}{x}+\frac{1}{y}} & x > 0 \wedge y > 0 \\ \min(x,y) & \text{otherwise} \end{cases}$$

TeLEx is similar to Max, with the exception of the Peak and Expand transformations defined in the TeLEx project [21]. Since the goal of TeLEx is to learn tight specifications in the absence of negative examples, these transformations cause the semantics to score minimally satisfied specifications higher. The Peak transformation (P) is applied within the rectifier, causing the robustness score to increase sharply for small positive values close to zero. The Expand transformation (E) is a multiplicative coefficient applied to certain integrators used in $\mathcal{U}$ and depends on the length of the time horizon ($t_2 - t_1$). We note that the Contract function (C) from the original paper is unused in our instantiation, as we choose to define all our temporal operators in terms of $\mathcal{U}$. The transformations have adjustable parameters β and γ that can influence the degree to which the scores are transformed. The transformation functions are defined as follows:

$$P(x) = \frac{1}{x + e^{-\beta x}} - e^{-x} \qquad E(\gamma, t_1, t_2) = \frac{2}{1 + e^{-\gamma(t_2 - t_1 + 1)}}$$

We note that TeLEx semantics were designed for specification learning, rather than falsification. For our benchmark evaluation, we arbitrarily chose the values $\beta = 1$ and $\gamma = 0.01$.

Cumulative is based on the smooth cumulative robustness semantics of Haghighi, Mehdipour, Bartocci, and Belta [17], where the max function has been

"smoothened" in a way resembling LogSumExp: $\max_\beta(x, y) = \frac{1}{\beta}\ln(e^{\beta x} + e^{\beta y})$. The min function is defined as $\min_\beta(x, y) = -\max_\beta(-x, -y)$. β in this instantiation represents the degree to which these functions approximate max and min, with higher values having a "sharper" response that more closely resembles their non-smooth counterparts. These operators are then generalized to trace integrators which, unique to the semantics we considered from literature, integrate over time rather than simply choosing an extreme value. We note that this semantics is not sound; $\min_\beta$ tends to be an over-estimation, resulting in $\min_\beta(x, 0) > 0$ for some values of $x < 0$. We arbitrarily chose $\beta = 10$ for our evaluations but note that the operator remains unsound regardless of its particular value. This makes falsification more difficult, as true falsifying traces may be incorrectly assigned positive robustness values

The use of summation in this semantics' definition of $\mathcal{U}$ is intended to more strongly reward traces which satisfy a specification for longer periods of time, rather than considering only a maximally-satisfying point. We note that this semantics requires that both $\rho^+(\varphi \mathbf{U} \psi, \sigma, t_k)$ and $\rho^-(\varphi \mathbf{U} \psi, \sigma, t_k)$ employ summation despite the expected multiplication or min in the computation of ρ^-.

We also note that integrator functions can be mixed to form novel semantics, provided they are sound. **Sum-Product, Sum-Min, Max-Product** show the three other possible permutations of mixing max or min and summation or multiplication for the integrators. We note that Sum-Min bears some superficial resemblance to Cumulative semantics [17].

β, η, Γ, and Ξ are thought of as "optimistic" integrators, as they tend to favor robustness scores with larger magnitudes, and are usually instantiated with something resembling max or addition. Making these operators unnecessarily strong, with min or multiplication, preserves soundness but may make the semantics overly conservative, to the point where all scores are zero and thus no optimization progress can be made. **MinOnly** illustrates this, by replacing all non-rectifier operators with min.

All of the semantics instantiations we have introduced so far use either non-smooth rectifiers, or unsound, smooth approximations. We introduce **Smooth-Rect** semantics, which uses the Max semantics but with sound, smooth rectifiers R_+ and R_-:

$$R_+(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ xe^{-1/x} & \text{if } x > 0 \end{cases} \qquad R_-(x) = \begin{cases} 0 & \text{if } x \geq 0 \\ xe^{1/x} & \text{if } x < 0 \end{cases}$$

We create **Smooth-1** as a smooth alternative to Sum-Product, using the differentiable, sound rectifiers R_+ and R_-. Differentiability can be useful in conjunction with gradient-based solvers [21,24]. Although this should provide a sound differentiable semantics in theory, the use of products may lead to numerical overflow in practice.

Table 1: Semantics instantiations

	Smooth/ Sound	ν	μ	α	β
Max [11]	N/Y	$\max(\cdot, 0)$	$\min(\cdot, 0)$	min	max
Const [11]	N/Y	$\max(A \times \text{sgn}(\cdot), 0)$	$\min(A \times \text{sgn}(\cdot), 0)$	min	max
Add [11]	N/Y	$\max(\cdot, 0)$	$\min(\cdot, 0)$	Koen &+	Koen &+
TeLEx [21]	N/Y	$\max(P(\cdot), 0)$	$\min(P(\cdot), 0)$	min	max
Cumulative [17]	Y/N	$\max_\beta(\cdot, 0)$	$\min_\beta(\cdot, 0)$	$\min_\beta$	$\max_\beta$
Sum-Product	N/Y	$\max(\cdot, 0)$	$\min(\cdot, 0)$	$\times$	$+$
Sum-Min	N/Y	$\max(\cdot, 0)$	$\min(\cdot, 0)$	min	$+$
Max-Product	N/Y	$\max(\cdot, 0)$	$\min(\cdot, 0)$	$\times$	max
MinOnly	N/Y	$\max(\cdot, 0)$	$\min(\cdot, 0)$	min	min
SmoothRect	N/Y	$R_+(\cdot)$	$R_-(\cdot)$	min	max
Smooth-1	Y/Y	$R_+(\cdot)$	$R_-(\cdot)$	$\times$	$+$

	ζ	η	Γ	Δ	Θ	Ξ
Max [11]	min	max	max	min	min	max
Const [11]	min	max	max	min	min	max
Add [11]	min	max	max	min	min	max
TeLEx [21]	min	max	$E \cdot \max$	min	min	$E \cdot \max$
Cumulative [17]	$\min_\beta$	$\max_\beta$	$\sum$	$\min_\beta$	$\sum$	$\max_\beta$
Sum-Product	$\times$	$+$	$\sum$	$\prod$	$\prod$	$\sum$
Sum-Min	min	$+$	$\sum$	min	min	$\sum$
Max-Product	$\times$	max	max	$\prod$	$\prod$	max
MinOnly	min	min	min	min	min	min
SmoothRect	min	max	max	min	min	max
Smooth-1	$\times$	$+$	$\sum$	$\prod$	$\prod$	$\sum$

4 Soundness of Semantics

Sound quantitative semantics ensures that no falsifying traces for a particular system and specification are incorrectly deemed safe with a positive robustness score. For falsification, soundness ensures that falsifying traces that exist in the search space can indeed be found. On the other hand, soundness in parameter or control synthesis ensures that unsafe controls are not admitted, or that they can at least be readily identified. The consequence of unsoundness in falsification is that unsafe traces may be "masked", making them more difficult or even impossible to find. While the absence of counterexamples in a falsification search cannot and should not imply safety, unsoundness ultimately introduces additional uncertainty that can confound the design of safe systems.

We now introduce novel, sufficient soundness conditions on the aforementioned functions. Recall that functions α, β, ζ, η, Γ, Δ, Θ and Ξ only operate on nonnegative values. To avoid ambiguity, we use $+$ and $\times$ to represent binary addition and multiplication, and $\sum$ and $\prod$ to represent their trace counterparts.

(1) If $x \leq 0$ then $\nu(x) = 0$. Equivalently, if $\nu(x) > 0$ then $x > 0$.
Example of possible ν: $\max(\cdot, 0)$.
(2) If $x \geq 0$ then $\mu(x) = 0$. Equivalently, if $\mu(x) < 0$ then $x < 0$.
Example of possible μ: $\min(\cdot, 0)$.
(3) If $x \geq 0$, $y \geq 0$, if $x = 0$ **or** $y = 0$ then $\alpha(x, y) = 0$. Equivalently, if $x \geq 0$, $y \geq 0$ and $\alpha(x, y) > 0$, then $x > 0$ **and** $y > 0$.
Examples of possible α: min, $\prod$ (discrete product).
(4) If $x \geq 0$, $y \geq 0$, if $x = 0$ **and** $y = 0$ then $\beta(x, y) = 0$. Equivalently, if $x \geq 0$, $y \geq 0$ and $\beta(x, y) > 0$, then $x > 0$ **or** $y > 0$.
Examples of possible β: max, sum $\sum$.
(5) If all $x_k = 0$, then $\left(\prod_k x_k\right) = 0$. Equivalently, if all $x_k \geq 0$ and $\left(\prod_k x_k\right) > 0$, then there exists k for which $x_k > 0$.
Examples of possible $\prod$: max, $\sum$ (discrete sum), $\int$ (integral).
(6) If there exists a k for which $x_k = 0$, then $\left(\bigtriangleup_k x_k\right) = 0$. Equivalently, if all $x_k \geq 0$ and $\left(\bigtriangleup_k x_k\right) > 0$, then all $x_k > 0$.
Examples of possible $\bigtriangleup$: min, $\prod$.
(7) ζ follows the requirements of α, and η the ones of β. Θ follows the requirements of $\bigtriangleup$, and Ξ the ones of $\prod$.

Using those conditions, we can now prove a novel generic soundness theorem:

Lemma 2. *Under conditions (1)-(7), if $\rho^+(\varphi, \sigma, t) > 0$ then $\sigma^t \vDash \varphi$, and if $\rho^-(\varphi, \sigma, t) < 0$ then $\sigma^t \nvDash \varphi$.*

Proof. The two statements are proved jointly by structural induction on φ, using the definition of φ given in Section 2.3 and applying conditions (1)-(7) as needed. Condition (1) on ν is used for the case $\rho^+(l(\sigma) \geq 0, \sigma, t)$ which is defined with ν; condition (2) on μ is similarly used for the case $\rho^-(l(\sigma) \geq 0, \sigma, t)$ which is defined with μ. Condition (3) on α is used for the $\wedge$ case on ρ^+, and condition (4) on β is used for the $\wedge$ case on ρ^-. Conditions (5)-(7) are used for the $\mathbf{U}_I$ case on ρ^+, and condition (7) is used for the $\mathbf{U}_I$ on ρ^-. A detailed proof is available in the extended version of the paper.

Theorem 2. *Under conditions (1)-(7), if $\rho(\varphi, \sigma, t) > 0$ then $\sigma^t \vDash \varphi$, and if $\rho(\varphi, \sigma, t) < 0$ then $\sigma^t \nvDash \varphi$.*

Proof. Since by definition $\rho(\varphi, \sigma, t) = \rho^+(\varphi, \sigma, t) + \rho^-(\varphi, \sigma, t)$ and by Lemma 1 $\rho^-(\varphi, \sigma, t) \leq 0$, if $\rho(\varphi, \sigma, t) > 0$ then $\rho^+(\varphi, \sigma, t) > 0$. Therefore by Lemma 2 we get $\sigma^t \vDash \varphi$. Symmetrically, since by Lemma 1 $\rho^+(\varphi, \sigma, t) \geq 0$, if $\rho(\varphi, \sigma, t) < 0$ then $\rho^-(\varphi, \sigma, t) < 0$. Therefore by Lemma 2, $\sigma^t \nvDash \varphi$.

Corollary 1. *Under conditions (1)-(7), we cannot have both $\rho^+(\varphi, \sigma, t) > 0$ and $\rho^-(\varphi, \sigma, t) < 0$. That is, if $\rho^+(\varphi, \sigma, t) > 0$ then $\rho^-(\varphi, \sigma, t) = 0$; and likewise if $\rho^-(\varphi, \sigma, t) < 0$ then $\rho^+(\varphi, \sigma, t) = 0$. In other words, $\rho^+(\varphi, \sigma, t)$ is the positive part of $\rho(\varphi, \sigma, t)$, i.e., $\rho^+(\varphi, \sigma, t) = \max(0, \rho(\varphi, \sigma, t))$; and symmetrically $\rho^-(\varphi, \sigma, t)$ is the negative part of $\rho(\varphi, \sigma, t)$, i.e., $\rho^-(\varphi, \sigma, t) = \min(0, \rho(\varphi, \sigma, t))$.*

Proof. If $\rho^+(\varphi, \sigma, t) > 0$ and $\rho^-(\varphi, \sigma, t) < 0$, then by Lemma 2, both $\sigma^t \vDash \varphi$ and $\sigma^t \nvDash \varphi$, which is not possible.

A counterexample to the soundness claimed in existing work

Our framework is particularly effective at ensuring that existing and future quantitative semantics are sound. As an example of this effectiveness, we have found what we believe to be an error in an existing paper describing cumulative quantitative semantics [17]. The error was found while trying to fit the semantics of this paper to our generalized semantics. Below we provide a counterexample to their theorem as well as proposed resolution.

The paper mentions proving the soundness of the semantics by structural induction on φ, proving both $\rho^+(\varphi, \sigma, t_k) > 0 \Rightarrow \sigma^{t_k} \vDash \varphi$ and $\rho^-(\varphi, \sigma, t_k) < 0 \Rightarrow \sigma^{t_k} \nvDash \varphi$ simultaneously. However after verification with that proof strategy, there seems to be one case that does not verify, namely the case $\rho^-(\psi \mathbf{U}_I \varphi, \sigma, t_k)$. We contacted the authors multiple times but were not able to get an answer to our technical questions.

A counterexample. The example takes a single variable x, with $\sigma^{t_0}(x) = 1$, $\sigma^{t_1}(x) = 3$, $\sigma^{t_2}(x) = -5$, and the formula $\varphi = \neg((x \geq 0)\mathbf{U}_{[0,2]}(x - 2 \geq 0))$. For that example, clearly $\sigma^{t_0} \vDash (x \geq 0)\mathbf{U}_{[0,2]}(x - 2 \geq 0)$ and thus $\sigma^{t_0} \nvDash \varphi$, and yet $\rho^+(\varphi, \sigma, t_0) = 8 > 0$. A complete derivation is available in the extended version of the paper.

A proposed fix. We believe, the issue stems from the combination of the definitions of $\rho^+(\neg\varphi, \sigma, t_k)$ and $\rho^-(\psi \mathbf{U}_I \varphi, \sigma, t_k)$. A solution might be to maintain the definition of $\rho^+(\psi \mathbf{U}_I \varphi, \sigma, t_k)$, but change the definition of $\rho^-(\psi \mathbf{U}_I \varphi, \sigma, t_k)$ to:

$$\rho^-(\psi \mathbf{U}_I \varphi, \sigma, t_k) = \max_{k' \in I}(\min\{\rho^-(\varphi, \sigma, t_{k+k'}), \min_{k'' \in [k, k+k']} \rho^-(\psi, \sigma, t_{k''})\})$$

This uses the standard definition of quantitative semantics, but for ρ^-. Although not symmetric or smooth anymore, we believe that the two definitions are still compatible, and are able to prove soundness through our generalized framework.

5 Experimental Evaluation

Our experiments seek to answer three research questions:

RQ1: With all other solver details being equal, does a particular semantics consistently outperform the others in falsification?

RQ2: Does our generalization enable the creation of novel semantics that compare favorably to existing ones?

RQ3: Is the general framework able to replicate the results of existing benchmarking experiments?

We implemented the generalized semantics in MATLAB, by modifying the existing Breach toolkit [11] to calculate the positive (ρ^+) and negative (ρ^-) parts of ρ separately for semantics that integrate via summing [17]. We then implemented our generalized semantics and the instantiations given in Table 1. The source code for our modifications can be found at https://github.com/jchen-cs/breach Apart from the ARCH 2020 replication experiment [12], all evaluations were carried out on two workstations with Intel Xeon E3-1245 v3 CPUs running MATLAB r2024a on Ubuntu 22.04.4. The evaluation of a given benchmark problem and its semantics was run entirely on a single system.

Benchmarks

To answer **RQ1** and **RQ2**, we evaluated our semantics instantiations on falsification benchmarks partially inspired by existing examples in literature. We ensured that each benchmark had a falsifying trace so that all problems were feasible. The evaluations were done using Breach's default Nelder-Mead optimizer. The *autotrans*, *AFC*, *Path*, and *Proj* benchmarks were limited to 500 objective function evaluations, while *Path2* was limited to 1000 to account for its complexity. The Automatic Transmission and Fuel Control benchmarks were tested with the solver's corner search phase disabled, since they could be trivially falsified by extreme values. We did not impose a time limit on the benchmarks. In addition to our main evaluation, we also replicated a subset of an existing paper's experimental evaluation [12].

Automatic Transmission This benchmark, originally from [18], simulates the drivetrain of a vehicle with an automatic transmission, and has been used in existing works such as S-TaLiRo [1] and others [4,3,31]. We treat the system's throttle input as a piecewise constant function of uniform intervals and parameterize amplitudes. The specification to falsify is that the vehicle speed never exceeds 100 mph. The maximum throttle value was constrained to 70% to minimize "trivial" falsifications in the quasi-random search phase. This increased the likelihood that falsifications occur during the optimization phase, which is reliant on performance differences between semantics.

Abstract Fuel Control The Abstract Fuel Controller was originally implemented in Simulink [22], and was included as a sample benchmark in Breach. The model simulates an engine controller that regulates the air/fuel ratio based on engine speed and throttle position. For this model, we use the specification defined in the Breach sample that requires the air/fuel ratio to be within a specific range for the duration of the simulation. The engine speed is kept constant throughout the simulation, while the throttle position is controlled by a single pulse generator, parameterized by the maximum pedal angle (amplitude) and

the duration for which that position is held (pulse period). As with the previous benchmark, the parameter space was constrained to reduce trivial falsifications.

Projectile Trajectory The Projectile benchmark simulates a point mass being launched into a ballistic trajectory with a given initial velocity and launch angle. Safety is defined as the projectile avoiding a small area some distance away from the origin. As a result, most combinations of velocity and angle are safe.

Dubins Path This benchmark simulates an autonomous vehicle with a Dubins steering model (such as a car or airplane) that travels straight and then turns at a constant angular velocity, all while avoiding a stationary obstacle. The adjustable parameters are the vehicle's transverse velocity (u_v), the time at which the turn occurs (t_{turn}), and the angular velocity of the turn (u_ω). The obstacle is small and relatively far away from the vehicle's starting position, so while a counterexample is feasible, it is unlikely to occur and thus makes falsification challenging.

Complex Dubins Path The Path Planning benchmark was inspired by control synthesis benchmarks in existing literature [31,17], which simulates an autonomous vehicle in a two-dimensional space along with an obstacle that should be avoided. As before, we use a Dubins vehicle model with controllable transverse and angular velocities. Unlike the previous benchmark, this scenario uses discretized dynamics and treats the inputs as piecewise constant functions with uniform intervals. We set the discretization interval of the control function to 0.1s, meaning that the 1.1s simulation provides eleven possible transverse and angular commands within the simulation timespan. The safety specification requires that the vehicle never enter the unsafe region at any point during the simulation. The dynamics are governed by the following equations that are updated every $\Delta t = 0.1$s:

$$\begin{aligned} x[k+1] &= x[k] + v[k]\cos(\theta)\Delta t \\ y[k+1] &= y[k] + v[k]\sin(\theta)\Delta t \\ \theta[k+1] &= \theta[k] + \omega[k]\Delta t \end{aligned}$$

where k refers to the state for an arbitrary time-step and $k+1$ the next state.

ARCH 2020 Replication To answer **RQ3** and provide additional insights on **RQ2**, we replicated a portion of the experimental evaluation conducted in an existing paper [12], but with our modified version of Breach and, in the interest of time, a subset of the aforementioned semantics. We limited our evaluation to the automatic transmission benchmarks (AT), and used the same propositions and naming convention as existing literature [13]. In addition to the max, additive, and constant semantics in the original evaluation, we also included TeLEx-like,

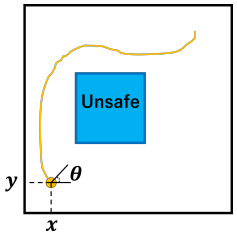

Fig. 1: Path planning model diagram

Sum-Product, and Sum-Min. As before, we used Breach with the built-in Nelder-Mead solver. We ran each combination of semantics and benchmark 10 times and set a maximum of 300 objective function calls. These experiments were performed on a 2017 iMac Pro with an 18-core 2.3 GHz Intel Xeon W processor and 128 GB of 2666 MHz DDR4 memory, running MATLAB r2024a on macOS 13.4.1.

6 Results

We performed falsification on each of the benchmarks, and for each semantics recorded the number of objective function evaluations until first falsfication (Table 2). In each benchmark column, the runs with the lowest number of iterations are displayed in **boldface**. Runs that did not find a counterexample within the allotted number of objective evaluations are marked DNF.

Among the benchmarks tested, we did not notice any clear patterns establishing the superiority of any particular semantics. However, the semantics that required fewer iterations to falsify a particular benchmark tended to be efficient in other benchmarks as well (**RQ1**). Constant semantics consistently required a large number of evaluations to falsify, which was consistent with both our own expectations and other published results [12]. Many of the top-performing semantics in each benchmark finished with similar or identical numbers of evaluations. The falsifications were all achieved well within the solver's optimization phase, rather than its random sampling phase, which suggests that differences in these semantics did not significantly affect falsification performance in our evaluation. The Smooth-1 semantics, while able to eventually falsify, required a relatively large number of evaluations; examining its traces revealed numerical overflow was occurring. Since the Nelder-Mead algorithm is gradient-free, we believe the benefits of a smooth semantics may be realized by comparison with gradient-based optimization. Nevertheless, the novel Sum-Min and SmoothRect semantics were comparable to existing semantics (**RQ2**)

The minimum-only semantics (minonly) did not find a falsifying trace in any benchmark. Examining the trace of its search showed that the robustness score returned by the semantics remained zero. Although the min-only semantics is

sound, it can be interpreted as overly conservative. The semantics based on [17] also did not find any falsifications. This could possibly be caused by the unsoundness of the semantics, as the reported robustness values do indeed change but remained positive. This would result in some true counterexamples being skipped in the search. Although it is possible to adjust the "softness" of the max and min functions used in these semantics [17], the adjustment would have to be tailored to a specific problem and the scale of its objective function values. The semantics inspired by TeLEx [21] found falsifying traces in all benchmarks, though often requiring more iterations. This is unsurprising as this semantics was optimized for policy learning rather than falsification. Unlike other instantiations of ν and μ, TeLEx features non-monotonic functions, which may complicate the search as robustness scores decrease in magnitude towards both zero and infinity. Product-based semantics (such as sum-product, max-product, and smooth1) tended to on occasion perform poorly. Examining their robustness values showed that these semantics frequently produced either very large or small values, and at times approached or exceeded the system's numerical limits. This may produce similar results to the constant semantics when large values saturate the robustness computation.

The results of the ARCH 2020 replication experiment are shown in Table 3. Each entry shows the number of successful falsifications out of 10, as well as the average number of objective function calls until successful falsification (parenthesized). The run with the most successful falsifications is displayed in **boldface**, with ties broken by the number of evaluations. Most of the AT benchmarks were falsified consistently by all semantics, and required on average a low number of evaluations until a falsifying trace is found. Compared to published results, Max, Const, and Add semantics generally performed similarly in relative performance in our replication experiment with similar trends in the mean evaluations until falsification. Therefore, we believe that, for **RQ3**, the semantics implemented in our generalization compare similarly to those from existing evaluation. In addition, sum-min semantics had the highest falsification rate of all semantics, and had the lowest average number of evaluations per successful falsification in 4 out of the 9 benchmarks. This indicates that, for the benchmarks we evaluated, this semantics is comparable to, and sometimes better than existing semantics from the literature (**RQ2**)

7 Conclusion and Future Work

The data we collected in our experimental evaluation does not appear to favor any particular semantics for all problems. This is to be expected given the diversity of problems and the relatively small dataset. Nevertheless, our generalization and implementation thereof provides a method to directly compare various semantics

Therefore, concurring with other evaluations [12], we believe that system designers employing falsification should try several quantitative semantics, or experiment to find semantics that perform best for their application. Using our

Table 2: Objective Evaluations until Falsification

	autotrans	AFC	Proj	Path	Path2
Max [11]	**65**	**43**	**26**	55	**352**
Const [11]	205	63	127	178	560
Add [11]	**65**	**43**	**26**	55	365
TeLEx [21]	205	**43**	348	358	**352**
Cumulative [17]	DNF	DNF	DNF	DNF	DNF
Sum-Product	397	63	359	**53**	560
Sum-Min	**65**	**43**	**26**	55	365
Max-Product	397	63	359	**53**	560
MinOnly	DNF	DNF	DNF	DNF	DNF
SmoothRect	**65**	**43**	**26**	55	**352**
Smooth-1	397	63	359	173	560

Table 3: ARCH Replication Experiment, Falsifications out of 10 (Mean evaluations per falsification)

	AT1	AT2	AT51	AT52	AT53
Max [11]	8 (151.4)	10 (11)	10 (34.1)	**10 (11.8)**	10 (16.1)
Const [11]	0 (-)	10 (16.8)	**10 (8.1)**	10 (16.8)	10 (8.8)
Add [11]	5 (156.2)	10 (15.2)	10 (36.4)	10 (20.3)	10 (16.3)
TeLEx [21]	0 (-)	10 (13.7)	10 (14.8)	10 (17.9)	10 (12.8)
Sum-Product	0 (-)	**10 (6.6)**	10 (16.2)	10 (16.9)	10 (7.3)
Sum-Min	**9 (173.6)**	10 (13.6)	10 (9.7)	10 (14)	**10 (6.2)**

	AT54	AT6a	AT6b	AT6c
Max [11]	10 (84.9)	10 (56)	9 (49.7)	10 (29.9)
Const [11]	10 (55)	9 (87.7)	**10 (39.9)**	10 (57)
Add [11]	10 (69.5)	**10 (20.8)**	10 (45.2)	10 (33.8)
TeLEx [21]	10 (97.2)	10 (62.8)	10 (70.7)	10 (56.2)
Sum-Product	10 (84.7)	8 (134.9)	7 (50.6)	10 (53.3)
Sum-Min	**10 (49.1)**	10 (55.2)	10 (55.5)	**10 (25.3)**

generalization, one may even develop bespoke semantics for a particular problem, with the assurance that its correctness can be easily checked using our soundness conditions. Unlike in existing work, this is facilitated by the general framework as it allows direct comparisons of semantics in the same environment, leaving other implementation details constant.

Our current experimental evaluation only examines the use of quantitative semantics in falsification tasks. Nevertheless, a generalized semantics framework can be employed in other application areas, such as control synthesis [7,17] or policy learning [21]. As with falsification, it is likely that performance will depend on choice of semantics or properties inherent to particular instantiations. It is possible that semantics that perform well in falsification are not necessarily optimal choices for other tasks. Additional experiments using benchmarks representative of these classes of problems will be necessary.

Our falsification experiments can also be expanded somewhat by examining other solvers or optimization techniques. Presently, we use Breach's Nelder-Mead optimizer, a gradient-free method, since many of the evaluated semantics are not differentiable. Other works have proposed using gradient-based solvers, paired with differentiable semantics, to potentially speed up the search process [21,24]. We believe this will allow us to, for instance, evaluate the performance improvement of pairing smooth semantics with a gradient-based solver. In addition to soundness and differentiability, we may consider other "meta-properties" of semantics instantiations and their effects on performance. For example, monotonicity may be suggestive of efficiency when paired with a gradient-following solver, while kernel size may be proportional to falsification difficulty

By developing a generalization of quantitative semantics, we not only enable easier direct comparisons of existing and novel quantitative semantics, but also provide a way to express useful meta-properties about them, such as soundness. We believe that by enabling direct comparisons of quantitative semantics, our framework will enable system designers to make informed decisions about their choice of semantics in optimization-based falsification.

Acknowledgements

We thank Koen Claessen, Pierre-Loïc Garoche, Elias Ghisellini, Umut Kurnaz and Marc Pouzet for valuable discussions on earlier versions of this work. We also thank Johan Lidén Eddeland, Sajed Miremadi, and Knut Åkesson for providing us the source code for their ARCH 2020 evaluation, and the anonymous reviewers for their feedback. This research was supported in part by National Science Foundation grants CCF-2348706, CCF-2426474, and CCF-2319572.

References

1. Abbas, H., Fainekos, G., Sankaranarayanan, S., Ivančić, F., Gupta, A.: Probabilistic temporal logic falsification of cyber-physical systems. ACM Trans. Embed. Comput. Syst. **12**(2s) (may 2013). https://doi.org/10.1145/2465787.2465797, https://doi.org/10.1145/2465787.2465797
2. Abbas, H., Winn, A., Fainekos, G., Julius, A.A.: Functional gradient descent method for Metric Temporal Logic specifications. In: 2014 American Control Conference. pp. 2312–2317. IEEE (2014). https://doi.org/10.1109/ACC.2014.6859453
3. Akazaki, T., Hasuo, I.: Time robustness in mtl and expressivity in hybrid system falsification. In: International Conference on Computer Aided Verification. pp. 356–374. Springer (2015)
4. Annpureddy, Y., Liu, C., Fainekos, G., Sankaranarayanan, S.: S-taliro: A tool for temporal logic falsification for hybrid systems. In: Abdulla, P.A., Leino, K.R.M. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 254–257. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
5. Bartocci, E., Bortolussi, L., Loreti, M., Nenzi, L.: Monitoring mobile and spatially distributed cyber-physical systems. In: Proceedings of the 15th ACM-IEEE International Conference on Formal Methods and Models for System Design. pp. 146–155. ACM, Vienna Austria (Sep 2017). https://doi.org/10.1145/3127041.3127050, https://dl.acm.org/doi/10.1145/3127041.3127050
6. Bartocci, E., Deshmukh, J., Donzé, A., Fainekos, G., Maler, O., Ničković, D., Sankaranarayanan, S.: Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications. In: Bartocci, E., Falcone, Y. (eds.) Lectures on Runtime Verification: Introductory and Advanced Topics, LNCS, vol. 10457, pp. 135–175. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75632-5_5
7. Cardona, G.A., Kamale, D., Vasile, C.I.: Mixed Integer Linear Programming Approach for Control Synthesis with Weighted Signal Temporal Logic. In: Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control. pp. 1–12. ACM, San Antonio TX USA (May 2023). https://doi.org/10.1145/3575870.3587120, https://dl.acm.org/doi/10.1145/3575870.3587120
8. Chattopadhyay, A., Mamouras, K.: A verified online monitor for metric temporal logic with quantitative semantics. In: Deshmukh, J., Ničković, D. (eds.) RV 2020. LNCS, vol. 12399, pp. 383–403. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60508-7_21
9. Claessen, K., Smallbone, N., Eddeland, J., Ramezani, Z., Åkesson, K.: Using Valued Booleans to Find Simpler Counterexamples in Random Testing of Cyber-Physical Systems. IFAC-PapersOnLine **51**(7), 408–415 (2018). https://doi.org/10.1016/j.ifacol.2018.06.333, https://linkinghub.elsevier.com/retrieve/pii/S2405896318306633
10. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: International Conference on Formal Modeling and Analysis of Timed Systems. pp. 92–106. Springer (2010)
11. Donzé, A.: Breach, A Toolbox for Verification and Parameter Synthesis of Hybrid Systems. In: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Touili, T., Cook, B., Jackson, P. (eds.) Computer Aided Verification, vol. 6174, pp. 167–170. Springer Berlin

Heidelberg, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14295-6_17, http://link.springer.com/10.1007/978-3-642-14295-6_17, series Title: Lecture Notes in Computer Science
12. Eddeland, J.L., Miremadi, S., Åkesson, K.: Evaluating optimization solvers and robust semantics for simulation-based falsification. In: Frehse, G., Althoff, M. (eds.) 7th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH20). EPiC Series in Computing, vol. 74, pp. 259–266. EasyChair (2020). https://doi.org/10.29007/f4vs
13. Ernst, G., Arcaini, P., Donze, A., Fainekos, G., Mathesen, L., Pedrielli, G., Yaghoubi, S., Yamagata, Y., Zhang, Z.: ARCH-COMP 2019 Category Report: Falsification. EPiC Series in Computing, vol. 61, pp. 129–140 (2019)
14. Faella, M., Legay, A., Stoelinga, M.: Model Checking Quantitative Linear Time Logic. Electronic Notes in Theoretical Computer Science **220**(3), 61–77 (Dec 2008). https://doi.org/10.1016/j.entcs.2008.11.019, https://linkinghub.elsevier.com/retrieve/pii/S1571066108004568
15. Fainekos, G.E., Pappas, G.J.: Robustness of temporal logic specifications for continuous-time signals. Theoretical Computer Science **410**(42), 4262–4291 (2009)
16. Gilpin, Y., Kurtz, V., Lin, H.: A smooth robustness measure of signal temporal logic for symbolic control. IEEE Control Systems Letters **5**(1), 241–246 (2020)
17. Haghighi, I., Mehdipour, N., Bartocci, E., Belta, C.: Control from signal temporal logic specifications with smooth cumulative quantitative semantics. In: 2019 IEEE 58th Conference on Decision and Control (CDC). pp. 4361–4366. IEEE (2019)
18. Hoxha, B., Abbas, H., Fainekos, G.: Benchmarks for temporal logic requirements for automotive systems. In: ARCH14-15. 1st and 2nd International Workshop on Applied veRification for Continuous and Hybrid Systems. pp. 25–18 (2015). https://doi.org/10.29007/xwrs, https://easychair.org/publications/paper/4bfq
19. Jakšić, S., Bartocci, E., Grosu, R., Ničković, D.: An Algebraic Framework for Runtime Verification. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **37**(11), 2233–2243 (Nov 2018). https://doi.org/10.1109/TCAD.2018.2858460, https://ieeexplore.ieee.org/document/8493500/
20. Jeannin, J.B., Chen, J., de Mendonça, J.L.V., Mamouras, K.: Work-in-progress: Towards a theory of robust quantitative semantics for signal temporal logic. In: 2022 International Conference on Embedded Software (EMSOFT). pp. 11–12. IEEE (2022)
21. Jha, S., Tiwari, A., Seshia, S.A., Sahai, T., Shankar, N.: TeLEx: learning signal temporal logic from positive examples using tightness metric. Formal Methods in System Design **54**(3), 364–387 (Nov 2019). https://doi.org/10.1007/s10703-019-00332-1, http://link.springer.com/10.1007/s10703-019-00332-1
22. Jin, X., Deshmukh, J.V., Kapinski, J., Ueda, K., Butts, K.: Powertrain control verification benchmark. In: Proceedings of the 17th international conference on Hybrid systems: computation and control. pp. 253–262. HSCC '14, Association for Computing Machinery, New York, NY, USA (Apr 2014). https://doi.org/10.1145/2562059.2562140, https://doi.org/10.1145/2562059.2562140
23. Koymans, R.: Specifying real-time properties with metric temporal logic. Real-Time Systems **2**(4), 255–299 (1990). https://doi.org/10.1007/BF01995674
24. Leung, K., Aréchiga, N., Pavone, M.: Backpropagation through signal temporal logic specifications: Infusing logical structure into gradient-based methods. The International Journal of Robotics Research **42**(6), 356–370 (May

2023). https://doi.org/10.1177/02783649221082115, https://doi.org/10.1177/02783649221082115, publisher: SAGE Publications Ltd STM

25. Lidén Eddeland, J., Claessen, K., Smallbone, N., Ramezani, Z., Miremadi, S., Åkesson, K.: Enhancing Temporal Logic Falsification With Specification Transformation and Valued Booleans. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **39**(12), 5247–5260 (Dec 2020). https://doi.org/10.1109/TCAD.2020.2966480, https://ieeexplore.ieee.org/document/8957695
26. Lindemann, L., Dimarogonas, D.V.: Robust control for signal temporal logic specifications using discrete average space robustness. Automatica **101**, 377–387 (2019). https://doi.org/https://doi.org/10.1016/j.automatica.2018.12.022, https://www.sciencedirect.com/science/article/pii/S0005109818306289
27. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: Lakhnech, Y., Yovine, S. (eds.) FTRTFT/FORMATS 2004. LNCS, vol. 3253, pp. 152–166. Springer, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30206-3_12
28. Mamouras, K., Chattopadhyay, A., Wang, Z.: Algebraic quantitative semantics for efficient online temporal monitoring. In: Groote, J.F., Larsen, K.G. (eds.) TACAS 2021. LNCS, vol. 12651, pp. 330–348. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72016-2_18
29. Mamouras, K., Chattopadhyay, A., Wang, Z.: A compositional framework for quantitative online monitoring over continuous-time signals. In: Feng, L., Fisman, D. (eds.) RV 2021. LNCS, vol. 12974, pp. 142–163. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88494-9_8
30. Mehdipour, N., Vasile, C.I., Belta, C.: Arithmetic-geometric mean robustness for control from signal temporal logic specifications. In: 2019 American Control Conference (ACC). pp. 1690–1695. IEEE (2019)
31. Mehdipour, N., Vasile, C.I., Belta, C.: Average-based robustness for continuous-time signal temporal logic (2019)
32. Nghiem, T., Sankaranarayanan, S., Fainekos, G., Ivancić, F., Gupta, A., Pappas, G.J.: Monte-Carlo techniques for falsification of temporal properties of non-linear hybrid systems. In: Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control. pp. 211–220. HSCC '10, ACM, New York, NY, USA (2010). https://doi.org/10.1145/1755952.1755983
33. Pant, Y.V., Abbas, H., Mangharam, R.: Smooth operator: Control using the smooth robustness of temporal logic. In: 2017 IEEE Conference on Control Technology and Applications (CCTA). pp. 1235–1240. IEEE (2017)
34. Sankaranarayanan, S., Fainekos, G.: Falsification of temporal properties of hybrid systems using the cross-entropy method. In: Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control. pp. 125–134. HSCC '12, ACM, New York, NY, USA (2012). https://doi.org/10.1145/2185632.2185653
35. Sato, S., Waga, M., Hasuo, I.: Constrained Optimization for Hybrid System Falsification and Application to Conjunctive Synthesis. IFAC-PapersOnLine **54**(5), 217–222 (2021). https://doi.org/10.1016/j.ifacol.2021.08.501, https://linkinghub.elsevier.com/retrieve/pii/S2405896321012763
36. Zhang, Z., Lyu, D., Arcaini, P., Ma, L., Hasuo, I., Zhao, J.: Effective Hybrid System Falsification Using Monte Carlo Tree Search Guided by QB-Robustness. In: Silva, A., Leino, K.R.M. (eds.) Computer Aided Verification, vol. 12759, pp. 595–618.

Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-81685-8_29, https://link.springer.com/10.1007/978-3-030-81685-8_29, series Title: Lecture Notes in Computer Science