On the Quantitative Semantics of Regular Expressions over Real-Valued Signals*

Alexey Bakhirkin[1], Thomas Ferrère[2], Oded Maler[1], and Dogan Ulus[1]

[1] Université Grenoble-Alpes, VERIMAG, F-38000 Grenoble, France
CNRS, VERIMAG, F-38000 Grenoble, France
[2] IST Austria, Am Campus 1, 3400 Klosterneuburg, Austria

Abstract. Signal regular expressions can specify sequential properties of real-valued signals based on threshold conditions, regular operations, and duration constraints. In this paper we endow them with a quantitative semantics which indicates how robustly a signal matches or does not match a given expression. First, we show that this semantics is a safe approximation of a distance between the signal and the language defined by the expression. Then, we consider the robust matching problem, that is, computing the quantitative semantics of every segment of a given signal relative to an expression. We present an algorithm that solves this problem for piecewise-constant and piecewise-linear signals and show that for such signals the robustness map is a piecewise-linear function. The availability of an indicator describing how robustly a signal segment matches some regular pattern provides a general framework for quantitative monitoring of cyber-physical systems.

1 Introduction

Regular expressions (RE) are among the cornerstones of computer science, being one of several formalism that can express sets of sequences (languages) acceptable by finite-state automata. In addition to their application in domains such as lexical analysis and pattern matching, regular expression are used in verification as a specification formalism to express correct or erroneous behaviors of reactive systems. In this context they are used along with another popular specification formalism, linear-time temporal logic (LTL) [27] that can express (star-free) regular languages in a different and complementary style. For both formalisms, the commonly-used semantics consists of discrete-time sequences often defined over finite small alphabets without a rich structure.

Over the years several extensions related to these two aspects, namely, discrete time and discrete non-numerical values, have been pursued in various contexts. To model real-time systems, finite automata have been augmented with continuously-evolving clocks resulting in timed automata [2] that can generate and accept sets of timed behaviors consisting of Boolean signals or time-event sequences, where events and state

* This research was supported in part by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement nr. 306595 "STATOR", and by the Austrian Science Fund (FWF) under grants S11402-N23 (RiSE/SHiNE) and Z211-N23 (Wittgenstein Award).

transitions are embedded in the dense real-time axis, not forced to occur at pre-specified sampling points or clock ticks. On the specification side, temporal logics have been extended with real-time constructs resulting in logics such as metric temporal logic (MTL) [22] and its decidable fragment MITL [3]. These logics can express quantitative timing properties such as bounds on the temporal distance between two events. Likewise, timed regular expressions have been defined and one of their variants has been proved to be expressively equivalent to timed automata [6,5].

In terms of alphabets, recent years saw a growing interest in large or infinite alphabets taken from richer domains such as $\mathbb{N}$ or $\mathbb{R}$, admitting order and arithmetic operations. Such languages are accepted by symbolic automata [33] whose transitions are labeled by predicates such as inequalities. Various questions, such as minimization [10], closure under various operations and learnability [25] have been studied in this context. Temporal logics over sequences of numbers [28] and first-order temporal logics in general [9], as well as regular expressions over quantitative domains [4] have also been defined and investigated.

The starting point of this work is *signal temporal logic* (STL) [23,24], which combines the dense time of MTL with predicates over real-valued variables. As such it can be used to express properties of continuous and mixed signals resulting from the simulation (or measurement) of continuous and hybrid systems such as analog circuits or cyber-physical control systems. Given a simulation trace w and an STL formula φ, simple and efficient algorithms [23], linear in the length of the trace, can check whether w satisfies φ and liberate users from the tedious and error-prone task of evaluating such traces manually. These algorithms have been implemented in tools such as AMT [26] and Breach [14] and have been applied to case-studies in domains ranging from control systems and robotics, via electronic circuits to systems biology.

Satisfaction (or membership) is traditionally a yes/no matter and it cannot distinguish between robust and non-robust satisfaction, a meaningful issue in numerical domains. To take a simple example, the requirement that some variable x is always positive is equally satisfied by safe behaviors being all the time far above zero and more dangerous and edgy ones that approach zero but do not cross it. To capture this distinction, quantitative semantics have been proposed for various temporal logics [17,28,15] including STL along with efficient algorithms to compute it [16]. In a nutshell, with every STL formula φ and signal w, a real-valued robustness measure $\rho = \rho(\varphi, w)$ is associated, admitting the following two important properties:

1. The robustness ρ is positive if w satisfies φ, negative if w violates φ;
2. The φ-satisfaction of a signal w', whose maximal pointwise distance from w is smaller than ρ, is equal to that of w.

The inductive definition of the quantitative semantics of STL is isomorphic to the standard definition of the qualitative semantics. The semantics of atomic predicates such as $x \geq 0$ at a point t is defined as $x[t]$. Boolean operations $(\wedge, \vee, \neg)$ are interpreted in the algebra $(\min, \max, -)$, and temporal operators are interpreted as $\min$ or $\max$ over time windows according to the disjunctive/conjunctive nature of the operator.

In this paper we define and compute such a quantitative semantics for *signal regular expressions* (SRE), which are timed regular expressions that use numerical predicates as atoms. Our semantics satisfies the same two important properties stated above. For regular expressions, due to the special nature of concatenation which requires to check all possible factorizations of a sequence or signal, it is natural to solve the more general pattern matching problem: given an expression φ and a signal w, find the set of all segments $w[t,t')$ that satisfy (match) φ. The set of all segments of a signal w is captured by the triangle $T_w = \{(t,t') \in \mathbb{T}^2 \mid 0 \leq t \leq t' \leq |w|\}$. The segments that satisfy φ define a subset of T_w that we call the *match set* of φ in w and denote by $\mathcal{M}(\varphi, w)$. In [32] it was proven that for every Boolean signal of bounded variability, the match set is a finite union of zones that can be computed by induction on the structure of the expression. The analogous problem for the quantitative semantics is to compute the robust satisfaction degree ρ of φ w.r.t. w for every segment $(t,t') \in T_w$. This is the problem we solve in this paper.

2 Signal Regular Expressions

In this section, we introduce signal regular expressions (SRE), recall their qualitative semantics and introduce a quantitative semantics. Signal regular expressions are an adaptation of the timed regular expressions (TRE) of [6,5] designed to deal with real-valued, rather than Boolean signals. They are built from atomic constraints (e.g., $x \geq 2$ specifies a signal segment where x is above 2), standard regular operations, and *duration constraints*. For an expression φ and an interval I, the duration constraint $\langle\varphi\rangle_I$ specifies that the duration of the signal segment that matches φ should be within I.

Signals. A *signal* is a function $w : \mathbb{T} \to \mathbb{R}^n$ where $\mathbb{T} = [s, s')$ is a bounded interval of $\mathbb{R}_{\geq 0}$, called the *temporal domain*. We usually take $s = 0$. The length of signal w is given by $|w| = s' - s$. By $w[t] \in \mathbb{R}^n$, we denote the value of w at time $t \in \mathbb{T}$. By $w[t,t')$, we denote the signal that has the temporal domain $[t,t') \subseteq [s,s')$ and agrees with w on $[t,t')$. We call such a signal a *factor* of w. Signal values are accessed by variables from the set $\mathbb{X} = \{x_1, \ldots x_n\}$. For a variable $x \in \mathbb{X}$ we denote by $w_x : \mathbb{T} \to \mathbb{R}$ the projection of w on x. We say that w_x is a *component* of w.

Definition 1 (Syntax of SRE). *SREs are formed according to the grammar:*

$$\varphi ::= \varnothing \mid \epsilon \mid x \geq c \mid x \leq c \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \cdot \varphi \mid \varphi^* \mid \langle\varphi\rangle_I$$

where $x \in \mathbb{X}$, $c \in \mathbb{R}$, and I is an interval of $\mathbb{R}$ with integer bounds.

As standard, we write iterated concatenation of an expression φ using power notation: $\varphi^0 = \epsilon$, and $\varphi^k = \varphi^{k-1} \cdot \varphi$ for an integer $k > 0$.

The *qualitative semantics* of expression φ w.r.t. a signal w is given by the function μ that returns a Boolean value (in $\{0,1\}$), indicating whether w *matches* φ.

Definition 2 (Qualitative Semantics of SRE). *The semantics $\mu(\varphi, w)$ of expression φ w.r.t a signal w is defined inductively as follows:*

$$\begin{aligned}
\mu(\varnothing, w) = 1 &\quad\Leftrightarrow\quad \bot \\
\mu(\epsilon, w) = 1 &\quad\Leftrightarrow\quad |w| = 0 \\
\mu(x \geq c, w) = 1 &\quad\Leftrightarrow\quad |w| > 0 \textit{ and } \forall t \in [0, |w|),\ w_x[t] \geq c \\
\mu(x \leq c, w) = 1 &\quad\Leftrightarrow\quad |w| > 0 \textit{ and } \forall t \in [0, |w|),\ w_x[t] \leq c
\end{aligned}$$

$$\begin{array}{rcl}
\mu(\varphi \vee \psi, w) = 1 & \Leftrightarrow & \mu(\varphi, w) = 1 \textit{ or } \mu(\psi, w) = 1 \\
\mu(\varphi \wedge \psi, w) = 1 & \Leftrightarrow & \mu(\varphi, w) = 1 \textit{ and } \mu(\psi, w) = 1 \\
\mu(\varphi \cdot \psi, w) = 1 & \Leftrightarrow & \exists uv = w,\, \mu(\varphi, u) = 1 \textit{ and } \mu(\psi, v) = 1 \\
\mu(\varphi^*, w) = 1 & \Leftrightarrow & \exists k \geq 0,\, \mu(\varphi^k, w) = 1 \\
\mu(\langle\varphi\rangle_I, w) = 1 & \Leftrightarrow & \mu(\varphi, w) = 1 \textit{ and } |w| \in I
\end{array}$$

where uv denotes the concatenation of signals u and v.

The set of signals that matches an expression φ is called the *language* of φ; $\mathcal{L}(\varphi) = \{w \mid \mu(\varphi, w) = 1\}$.

We introduce a *quantitative* semantics of an expression φ w.r.t. a signal w, given by the function ρ that returns a real value (in $\mathbb{R} \cup \pm\infty$) and indicates how robustly w matches φ. We call it the *robustness* of φ w.r.t. w (or the robustness of w w.r.t. φ).

Definition 3 (Quantitative Semantics of SRE). *The robustness $\rho(\varphi, w)$ of an expression φ w.r.t. a signal w is defined inductively as follows:*

$$\begin{aligned}
\rho(\varnothing, w) &= -\infty \\
\rho(\epsilon, w) &= \begin{cases} +\infty, & \textit{if } |w| = 0 \\ -\infty, & \textit{otherwise} \end{cases} \\
\rho(x \geq c, w) &= \begin{cases} \inf_{t\in[0,|w|)} w_x[t] - c & \textit{if } |w| > 0 \\ -\infty & \textit{otherwise} \end{cases} \\
\rho(x \leq c, w) &= \begin{cases} \inf_{t\in[0,|w|)} c - w_x[t] & \textit{if } |w| > 0 \\ -\infty & \textit{otherwise} \end{cases} \\
\rho(\varphi \vee \psi, w) &= \max\{\rho(\varphi, w), \rho(\psi, w)\} \\
\rho(\varphi \wedge \psi, w) &= \min\{\rho(\varphi, w), \rho(\psi, w)\} \\
\rho(\varphi \cdot \psi, w) &= \sup_{uv=w} \min\{\rho(\varphi, u), \rho(\psi, v)\} \\
\rho(\varphi^*, w) &= \sup_{k\geq 0} \rho(\varphi^k, w) \\
\rho(\langle\varphi\rangle_I, w) &= \begin{cases} \rho(\varphi, w), & \textit{if } |w| \in I \\ -\infty, & \textit{otherwise} \end{cases}
\end{aligned}$$

where $\sup \emptyset = \inf \mathbb{R} = -\infty$ *and* $\inf \emptyset = \sup \mathbb{R} = +\infty$.

In what follows, we can assume without loss of generality that atomic expressions are of the form $x \geq 0$. Given a constraint $x \geq c$ (resp. $x \leq c$), we can replace it by the constraint $y \geq 0$, where y is a fresh variable, and the projection of w on y is defined as $w_y[t] = w_x[t] - c$ (resp. $c - w_x[t]$). This replacement preserves the quantitative and qualitative semantics, as well as the assumptions that we make later (e.g., the signal being piecewise-constant or continuous piecewise-linear, etc).

3 Properties of the Quantitative Semantics

In this section, we present two important properties of our semantics. First, we relate the qualitative and quantitative semantics based on a notion of a *distance*. Second, we show that the quantitative semantics of a Kleene star expression φ^* can be computed as a finite iteration of φ.

3.1 Robustness Estimate

We now introduce a metric on the signal space and then derive two notions of distance between signals and expressions.

Definition 4 (Signal Distance). *The (uniform norm) distance between signals v and w, denoted $\mathrm{d}(v, w)$, is defined by $\mathrm{d}(v, w) = \sup_{t \in \mathbb{T}} \max_{x \in \mathbb{X}} |v_x[t] - w_x[t]|$ if v and w have the same temporal domain $\mathbb{T}$, otherwise $\mathrm{d}(v, w) = +\infty$.*

Definition 5 (Expression Distance). *The* distance *from signal w to expression φ, denoted $\underline{\mathrm{d}}(\varphi, w)$, is defined by $\underline{\mathrm{d}}(\varphi, w) = \inf_{v \in \mathcal{L}(\varphi)} \mathrm{d}(v, w)$. The* co-distance *from signal w to expression φ, denoted $\overline{\mathrm{d}}(\varphi, w)$, is defined by $\overline{\mathrm{d}}(\varphi, w) = \inf_{v \notin \mathcal{L}(\varphi)} \mathrm{d}(v, w)$.*

Such (Hausdorff) distances indicate by how much w needs to be changed to satisfy or violate φ, respectively. The quantitative semantics of SRE has the following characteristics: on a given signal its sign indicates membership in the language of the expression, and its magnitude estimates the distance to language boundary.

Theorem 1 (Soundness). *Let φ be an expression, and w a signal. If $\rho(\varphi, w) > 0$ then $w \in \mathcal{L}(\varphi)$. Symmetrically if $\rho(\varphi, w) < 0$ then $w \notin \mathcal{L}(\varphi)$.*

Theorem 2 (Correctness). *Let φ be an expression, and v, w two signals. If $w \in \mathcal{L}(\varphi)$ and $\mathrm{d}(v, w) < \rho(\varphi, w)$ then $v \in \mathcal{L}(\varphi)$. Symmetrically if $w \notin \mathcal{L}(\varphi)$ and $\mathrm{d}(v, w) < -\rho(\varphi, w)$ then $v \notin \mathcal{L}(\varphi)$.*

These two characteristics are in fact direct corollaries of the following statement:

Lemma 1 (Distance Bounds). *Let φ be an expression and w a signal. Then, we have $-\underline{\mathrm{d}}(\varphi, w) \leq \rho(\varphi, w) \leq \overline{\mathrm{d}}(\varphi, w)$.*

Due to space constraints, we cannot give the full proof here. It proceeds by straightforward induction on the expression structure, using lattice properties of $\min$, $\max$, $\inf$ and $\sup$.

Theorems 1 and 2 derive from Lemma 1 as follows. Assume $\rho(\varphi, w) > 0$. We have $\overline{\mathrm{d}}(\varphi, w) > 0$ and thus $w \in \mathcal{L}(\varphi)$ by definition of $\overline{\mathrm{d}}(\varphi, w)$; the statement of Theorem 1 is proved. If $\mathrm{d}(v, w) < \rho(\varphi, w)$, then $\overline{\mathrm{d}}(\varphi, v) \geq \overline{\mathrm{d}}(\varphi, w) - \mathrm{d}(v, w) > 0$ and again $v \in \mathcal{L}(\varphi)$; the statement of Theorem 2 is proved. Symmetrical reasoning applies to the case $\rho(\varphi, w) < 0$; when $\rho(\varphi, w) = 0$ both statements hold vacuously true.

3.2 Star Boundedness

We now prove that for every signal w (from a broad class of *well-behaved* signals) there exists some index m such that $\rho(\varphi^*, w) = \rho(\bigvee_{n=0}^{m} \varphi^n, w)$. Intuitively, this is because for practical signals, there is a limit to how many non-redundant factors it can be partitioned into. In particular, if a factor v is *sufficiently short*, $\rho(\varphi^2, v) \leq \rho(\varphi, v)$, and v

does not need to be partitioned further during the computation of $\rho(\varphi^*, w)$. A similar result was obtained in [32] for the qualitative semantics of SRE, but for the quantitative case the proof is much more complicated since we do not assume the signals to be piecewise-constant.

Let us formalize what is a *sufficiently short* signal. For a pair of signals f and g over temporal domain $\mathbb{T}$, we write $f \leq g$ when $f[t] \leq g[t]$ for every $t \in \mathbb{T}$. A real signal f is *increasing* (*decreasing*) if $f[t] \leq f[t']$ (respectively $f[t] \geq f[t']$) for all $t < t' \in \mathbb{T}$. A signal that is increasing or decreasing is called *monotone*.

Definition 6 (Unitary Signal). *A signal w is unitary when $|w| < 1$, for every $x \in X$, w_x is monotone, and for every $x, y \in \mathbb{X}$, $w_x \leq w_y$ or $w_y \leq w_x$.*

Intuitively, a unitary signal is sufficiently short and need not be partitioned further during the computation of $\rho(\varphi^*, w)$. More formally, we have the following.

Lemma 2 (Square). *For a unitary signal u and an expression φ, $\rho(\varphi^2, u) \leq \rho(\varphi, u)$.*

Let us delay the proof of Lemma 2, and instead state and prove (using Lemma 2) an important result of this section.

Definition 7 (Well-behaved signal). *A signal w is* well-behaved *if there exists $k \in \mathbb{N}$ such that w can be factored into $w = u_1 u_2 \ldots u_k$ where every u_i for $i \in \{1, \ldots, k\}$ is unitary. The smallest such k is denoted $\kappa(w)$.*

Theorem 3 (Star is Bounded). *For a well-behaved signal w with $\kappa(w) = k$ we have $\rho(\varphi^*, w) = \rho(\bigvee_{n=0}^{2k+1} \varphi^n, w)$.*

Proof. Assume, in search of a contradiction, that $\rho(\varphi^{n+1}, w) > \rho(\varphi^n, w)$ for some $n > 2k + 1$. This means that there exists a decomposition $w = u_1 u_2 \ldots u_{n+1}$ such that for every $1 \leq i \leq n+1$, $\rho(\varphi, u_i) > \rho(\varphi^n, w)$. However $\kappa(w) = k$, so by pigeon hole principle there exists j such that the factor $u' = u_j u_{j+1}$ is unitary. From Lemma 2, $\rho(\varphi, u') \geq \rho(\varphi^2, u') \geq \min\{\rho(\varphi, u_j), \rho(\varphi, u_{j+1})\}$ and thus $\rho(\varphi, u') > \rho(\varphi^n, w)$. We obtain the decomposition $w = u_1 \ldots u_{j-1} u' u_{j+2} \ldots u_{n+1}$ with n factors, s.t. robustness of φ on every factor is greater than $\rho(\varphi^n, w)$, which contradicts the definition of $\rho(\varphi^n, w)$. Thus, it has to be that $\rho(\varphi^{n+1}, w) \leq \rho(\varphi^n, w)$ for every $n > 2k+1$. According to the semantics of Kleene star, $\rho(\varphi^*, w) = \sup_{n \geq 0} \rho(\varphi^n, w) = \max_{n=0}^{2k+1} \rho(\varphi^n, w) = \rho(\bigvee_{n=0}^{2k+1} \varphi^n, w)$. □

Due to space constraints, we cannot give the proof of Lemma 2 in full, but we guide the reader through its most important steps. Let us write $\varphi \approx_w \psi$ when $\rho(\varphi, v) = \rho(\psi, v)$ for every factor v of w; and $\varphi \preccurlyeq_w \psi$ when $\rho(\varphi, v) \leq \rho(\psi, v)$ for every factor v of w. It turns out, given a unitary signal w, we can always rewrite an expression φ into another expression γ, s.t. $\gamma \approx_w \varphi$ and for which we can prove $\gamma^2 \preccurlyeq_w \gamma$. Let us give an example of such a rewriting.

Example 1. Consider the unitary signal w defined on $[s_1, s_2)$ in Figure 1 and the expression $\varphi = \langle x \geq 0 \cdot z \geq 0 \rangle_{(0,2)} \vee y \geq 0$. Let v be an arbitrary factor $w[r_1, r_2)$. Observe that $\rho(x \geq 0 \cdot z \geq 0, v) = \sup_{r' \in (r_1, r_2)} \min\{\inf_{[r_1, r')} w_x, \inf_{[r', r_2)} w_z\} = \inf_{[r_1, r')} w_x$ (since w_x is increasing and everywhere below w_z). That is $x \geq 0 \cdot y \geq 0 \approx_w x \geq 0$.

Then, notice that the duration constraint has no influence on signals of length less than one. Finally, since w_y is pointwise below w_x, $x \geq 0 \vee y \geq 0 \;\approx_w\; x \geq 0$. Thus, $\varphi = \langle x \geq 0 \cdot z \geq 0 \rangle_{(0,2)} \vee y \geq 0 \approx_w x \geq 0$.

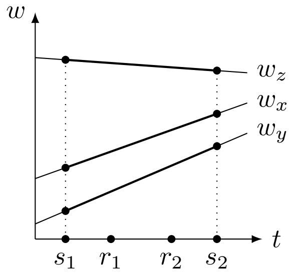

Fig. 1. A unitary signal w defined on $[s_1, s_2)$.

Example 2. Consider the same unitary signal, the expression $\varphi = x \geq 0 \cdot z \geq 0 \cdot y \geq 0$ and let v be an arbitrary factor $w[r_1, r_2)$. Observe that $\rho(\varphi, v) = \sup_{r',r'' \in (r_1,r_2)} \min\{\inf_{[r_1,r')} w_x, \inf_{[r',r'')} w_z, \inf_{[r'',r_2)} w_y\}$. To maximize the minimum, we want to move the time point r'' infinitely close to r_2 and obtain $\rho(\varphi, v) = \min\{\inf_v w_x, \sup_v w_y\}$ (since w_z is everywhere above w_x and w_y, the corresponding term is discarded). Similarly, we can show that $\rho(x \geq 0 \cdot y \geq 0, v) = \min\{\inf_v w_x, \sup_v w_y\}$ and thus $\varphi \approx_w x \geq 0 \cdot y \geq 0$.

In general, the robustness of an expression over a unitary signal *and over its every factor* is given by some min-max expression. We prefer to state this in a more algebraic way and say that for a given unitary signal, every expression can be rewritten to an equivalent *quadratic expression*.

Definition 8 (Monomial, Polynomial). *A* monomial expression *(of* degree *n) is of the form $\epsilon \cdot \alpha_1 \cdot \alpha_2 \cdots \alpha_n$, where every α_i is and atomic expression. A* polynomial expression *(of* degree *n) is of the form $\varnothing \vee \bigvee_{j=1}^{m} \beta_j$ where each β_j is a monomial expression (of degree at most n). A polynomial expression of degree at most 2 is called quadratic.*

Quadratic expressions are sufficient to represent the robustness of an arbitrary expressions over unitary signals. They are also necessary, as illustrated in Example 2, in the sense that atomic expressions cannot replace arbitrary expressions.

Proposition 1 (Quadratic Expressions). *For every expression φ and unitary signal u, there exists a quadratic expression γ such that $\varphi \approx_u \gamma$.*

The proof proceeds by structural induction, rewriting the expressions using Kleene algebra and lattice equivalences. Lemmas 3 and 4 state the key property of polynomial expressions relative to unitary signals. By proving them, we prove Lemma 2

Lemma 3 (Product of Monomials). *Let u be a unitary signal. For every pair of monomials β_1, β_2 there exists a monomial β_3 of degree at most 2 such that the following is true: $\beta_1 \cdot \beta_2 \approx_u \beta_3 \preccurlyeq_u \beta_1 \vee \beta_2$*

Proof idea. To find β_3, we want to find an atomic expression $y \geq 0$ appearing in $\beta_1 \cdot \beta_2$ such that y is minimal. Then if y is increasing, all factors in $\beta_1 \cdot \beta_2$ at the right of $y \geq 0$ can be ignored, and only the leftmost factor $x \geq 0$ is needed. We let $\beta_3 = x \geq 0 \cdot y \geq 0$, and check that monotonicity and ordering entail the equation of Lemma 3; the case where y is decreasing is symmetrical. □

Lemma 4 (Squaring Polynomials). *For every unitary signal u and polynomial expression γ we have $\gamma^2 \preccurlyeq_u \gamma$.*

Proof idea. For every pair of monomials β_1, β_2 appearing in γ, (via Lemma 3) we have $\beta_1 \cdot \beta_2 \preccurlyeq_u \beta_1 \vee \beta_2$. Distributing unions over concatenation we see that every monomial in γ^2 is dominated according to $\preccurlyeq_u$ by a monomial in γ. □

4 The Robust Matching Problem

Given an expression φ and a signal w, *robust matching* is the problem of computing the quantitative semantics of φ for every segment $w[t, t')$ of w.

Definition 9 (Robustness Map). *For an expression φ and a signal w, the robustness map is the function $(t, t') \mapsto \rho(\varphi, w[t, t'))$ that maps every t, t' (s.t. $0 \leq t \leq t' \leq |w|$) to the robustness of φ w.r.t w on $[t, t')$.*

It is convenient to represent the robustness map indirectly, using the following notion of *robustness support*.

Definition 10 (Robustness Support). *For a signal w and an expression φ, the robustness support is the set $\mathcal{R}(\varphi, w) = \{(t, t', r) \mid r < \rho(\varphi, w[t, t'))\}$*

The robustness map can be extracted from the robustness support by taking its pointwise supremum: $\rho(\varphi, w[t, t')) = \sup\{r \mid (t, t', r) \in \mathcal{R}(\varphi, w)\}$.

In what follows, we consider two classes of signals, continuous piecewise-linear and piecewise-constant. We show that for these signals, the robustness support can be represented as a finite union of polyhedra or *zones* respectively. As a result, the robustness map for these signals is respectively piecewise-linear or piecewise-constant.

Definition 11 (Zone). *A zone is a polyhedron formed by intersection of constraints of the form $x \bowtie c$ or $x - y \bowtie c$ where x and y are variables, c is a constant, and $\bowtie \in \{<, \leq, =, \geq, >\}$.*

Zones are often used in verification of timed systems. They admit efficient representation and computation via difference bound matrices [13].

Connection to Qualitative Matching. The *match set* $\mathcal{M}(\varphi, w)$ of an expression φ w.r.t. a signal w is the set of pairs (t, t') such that the factor $w[t, t')$ matches φ. That, is, $(t, t') \in \mathcal{M}(\varphi, w)$ iff $w[t, t') \in \mathcal{L}(\varphi)$. For a signal of bounded variability (i.e., when the truth value of every atomic proposition has finitely many switching points), this set was shown [32] to be computable and representable by a finite union of zones.

4.1 Finite Representation of Signals

Signals are typically represented by finitely many sampling points and interpolated between them. When computing robustness, we are interested in simple interpolation schemes that produce piecewise-constant or *continuous* piecewise-linear signals (our theoretical results are applicable to a larger class of signals). Formally, we can define

Definition 12 (Piecewise-Constant Signal). *A signal $w : \mathbb{T} \to \mathbb{R}^n$ is* piecewise-constant *if there exists a partition of $\mathbb{T}$ into a finite ordered sequence of* left-closed right-open *intervals $(J_1, \ldots, J_n)$, s.t. for every $i = 1 \ldots n$, the value of w on J_i is constant.*

Definition 13 (Piecewise-Linear Signal). *A signal $w : \mathbb{T} \to \mathbb{R}^n$ is piecewise-linear if there exists a partition of $\mathbb{T}$ into a finite sequence of left-closed right-open intervals $(J_i)_{1 \leq i \leq n}$, and there exist sequences of vectors $(a_i)_{1 \leq i \leq n}$ and $(b_i)_{1 \leq i \leq n}$ such that $w[t] = a_i t + b_i$ for every $t \in J_i$, for every $1 \leq i \leq n$.*

For both classes of signals, we call the endpoints of the intervals, $\inf J_i$ and $\sup J_i$, the *switching points*. When computing the robustness, we are only interested in piecewise-linear signals that are also continuous, i.e., where $a_k(\sup J_k)+b_k = a_{k+1}(\sup J_{k+1})+b_{k+1}$ for every two adjacent segments J_k, J_{k+1}.

The notion of a piecewise-constant or a piecewise-linear function can be extended to higher dimensions. In particular, for $\mathbb{T}^2 \to \mathbb{R}$, we can define:

Definition 14 (Piecewise-Constant and Piecewise-Linear Functions). *We say that a function $f : \mathbb{T}^2 \to \mathbb{R}$ is piecewise-constant if there exists a finite set of convex polyhedra $\{P_i\}_{1 \leq i \leq n}$ over t, t', s.t. $dom(f) = \bigcup_{i=1}^{n} P_i$, and on every P_i, f is constant. If on every P_i, f is linear in t, t', we say that f is piecewise-linear.*

Theorem 4 (Piecewise-Linear Decomposition). *For an expression φ and piecewise-constant signal w, the robustness map $(t, t') \mapsto \rho(\varphi, w[t, t'])$ is piecewise-constant. For a continuous piecewise-linear signal w, the robustness map is piecewise-linear.*

Section 5 gives a constructive proof of this.

As a final remark, we note that if we sample and interpolate an analog signal, for which we know the bound on its derivative, we can infer the bound on the pointwise distance between this analog signal and its piecewise representation. This number will also be the bound on the difference between the robustness of the original signal and the robustness of its piecewise representation.

5 Algorithms

In this section we present robust matching algorithms for piecewise-constant and continuous piecewise-linear signals. More specifically, our algorithms compute a polyhedral representation of the robustness support. For a signal w and an expression φ, by induction on the structure of φ, we compute a set $\mathcal{S}_\varphi$ of convex polyhedra (over t, t', and r) , whose union is $\mathcal{R}(\varphi, w)$. In particular, for a piecewise-constant signal, $\mathcal{S}_\varphi$ is a set of zones. We use the robustness support as an implicit representation of the robustness map: given a pair (t, t'), we take the maximum value of r over all (t, t', r) in the polyhedra in $\mathcal{S}_\varphi$. First, we show how to compute $\mathcal{S}_\varphi$ for the atomic propositions ($x \geq c$, $x \leq c$). Then, for the other operations, the robustness support is characterized by induction on the structure of the expression using basic operations on sets of convex polyhedra.

Atomic Propositions. Recall that we only need an algorithm to compute the robustness support for the basic atomic proposition $x \geq 0$. For a proposition such as $x \geq c$, we introduce an auxiliary signal component y defined as $w_y[t] = w_x[t] - c$ and then

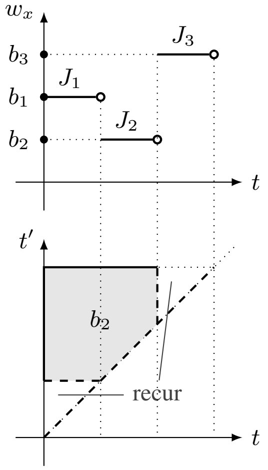

Fig. 2. A piecewise-constant signal component and a step of the algorithm for the expression $x \geq 0$

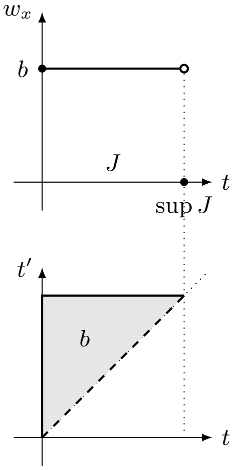

Fig. 3. A constant segment and its robustness for $x \geq 0$.

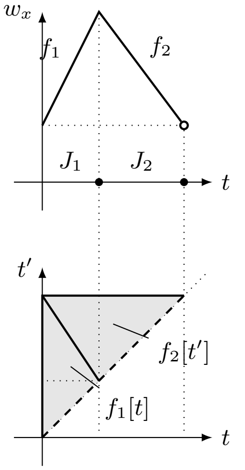

Fig. 4. A signal component consisting of two segments and its robustness for $x \geq 0$.

compute the support for $y \geq 0$. For the expression $x \geq 0$, robustness support can be characterized as follows

$$\mathcal{R}(x \geq 0, w) = \{(t, t', r) \mid r < \rho(x \geq 0, w, [t, t'))\} = \{(t, t', r) \mid r < \inf_{t'' \in (t,t')} w_x[t'']\}$$

That is, $\mathcal{S}_{x\geq 0}$ should be a set of polyhedra, where for every t and t', the value of r is bounded from above by the infimum of w_x on the interval (t, t'). For both piecewise-constant and continuous piecewise-linear signals we can compute it by induction on the *signal* structure, although in slightly different ways.

Piecewise-Constant Signals. Assume we are given a signal w and a finite sequence of intervals $(J_1, \ldots, J_n)$, s.t., on every J_i the signal is constant. Also assume that on an interval J_k, w_x reaches its global minimum b_k. Then, we immediately know that for every interval (t, t') that intersects with J_k, the robustness value is given by b_k. Based on this observation, we build the following recursive algorithm.

1. Given a sequence $(J_1, \ldots, J_n)$, find an interval J_k, where component w_x achieves its minimum value b_k.
2. Add to $\mathcal{S}_{x\geq 0}$ the zone $(\inf J_1 \leq t < \sup J_k \wedge \inf J_k < t' \leq \sup J_n \wedge t < t' \wedge r < b_k)$. It corresponds to the shaded area in Fig. 2.
3. If $k > 1$, apply the procedure recursively to the sequence $(J_1, \ldots, J_{k-1})$.
4. If $k < n$, apply the procedure recursively to the sequence $(J_{k+1}, \ldots, J_n)$.

In Fig. 2, we give an example of one step of this procedure.

The procedure produces one *zone* for every constant segment of the signal (every recursive call produces one zone and removes one segment from consideration). Also, every constraint on r is of the form $r < c$. This is important for Theorem 4.

Continuous Piecewise-Linear Signals. To simplify the presentation, we make two assumptions. First, we define the value of the signal at the right endpoint: $w[\sup \mathbb{T}] = \lim_{t \to \sup \mathbb{T}} w[t]$. Second, we assume that w_x reaches the minimum on both endpoints: $w_x[0] = w_x[\sup \mathbb{T}] = \min w_x$ (recursively, this will be preserved). We can always extend a signal to ensure this property, which saves us from extra case analysis. To get the idea of the algorithm, consider Fig. 5, which shows the general form of a signal with $n \geq 3$ segments. Let $x_{\min}$ be the minimal value of the signal on segments 2 to $n-1$, that is, excluding the first and the last segments, and let $t_{\min}$ be the corresponding time point. This information is sufficient for computing the robustness value for certain segments.

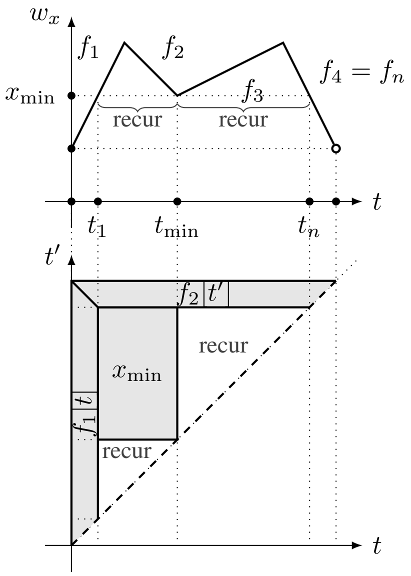

Fig. 5. A signal component with more than two segments and a step of the algorithm for the expression $x \geq 0$.

(1) In all segments that contain $t_{\min}$ and where the values in the first and last segments is above $x_{\min}$, the robustness value is the constant $x_{\min}$. This holds in the rectangle defined by $t \in [t_1, t_{\min}]$ and $t' \in [t_{\min}, t_n]$ where t_1 and t_n satisfy $f_1[t_1] = f_n[t_n] = x_{\min}$.
(2) When $t \leq t_1$ and $t' \leq t_n$, the robustness value is given by $f_1[t]$ and likewise when $t > t_1$ and $t' > t_n$, the value is given by $f_n[t']$.
(3) When $t \leq t_1$ and $t' > t_n$ the value is given by either $f_1[t]$ or $f_n[t']$ depending on a linear inequality based on their respective slopes.
(4) Other segments not covered by the above and where the minimal value is larger than $x_{\min}$, are subject to two recursive calls over the intervals $[t_1, t_{min}]$ and $[t_{min}, t_n]$.

More formally, Let w be a signal, x be a variable, and J be a time interval. We consider three cases, displayed in Fig. 3, 4, and 5.

Case 1: When w_x has one segment, because of our assumption, it is constant: $w_x[t] = b$, for $t \in J$, as shown in Fig. 3. Then for $t < t' \in J$, the robustness value is b. That is, we add to $\mathcal{S}_{x \geq 0}$ the zone $(t, t' \in J,\ t < t' \wedge r < b)$.

Case 2: When w_x consists of two segments, it always has a wedge-like shape as in Fig. 4. In this case we again can immediately produce the result. More formally, let J be split into J_1, J_2, s.t. $w_x[t] = a_1 t + b_1 = f_1[t]$ when $t \in J_1$, $a_2 t + b_2 = f_2[t]$ when $t \in J_2$, and (since w_x is continuous) $f_1[\sup J_1] = f_2[\inf J_2]$. Then, for $t < t' \in J_1 \cup J_2$, robustness is given by $\min\{\, f_1[t], f_2[t']\,\}$. Thus, we add to $\mathcal{S}_{x \geq 0}$ the polyhedron $(t, t' \in J_1 \cup J_2,\ t < t' \wedge r < a_1 t + b_1 \wedge r < a_2 t' + b_2)$.

Case 3: Now assume that w_x consists of three or more segments, i.e., J can be split into a sequence of adjacent intervals $J_1, \ldots, J_n$, s.t., for $t \in J_i$, $w_x[t] = f_i[t] = a_i t + b_i$.

We show an example in Fig 5. In this case, let us find $x'_{\min} = \min_{t \in J_2 \ldots J_{n-1}} w_x[t]$ and some $t_{\min}$, s.t. $w_x[t_{\min}] = x'_{\min}$ ($t_{\min}$ can always be found at a switching point). Then we find the time points $t_1 \in [\inf J_1, \sup J_1]$ and $t_n \in [\inf J_n, \sup J_n]$, s.t. $f_1[t_1] = f_n[t_n] = x'_{\min}$. Now we consider the following sub-cases, based on where t and t' lie w.r.t the time points $\inf J_1, t_1, t_{\min}, t_n$, and $\sup J_n$.

1. When $t \in [t_1, t_{\min}]$ and $t' \in [t_{\min}, t_n]$, we know that the robustness value should be $x'_{\min}$. Thus, we add to $\mathcal{S}_{x \geq 0}$ the polyhedron $(t \in [t_1, t_{\min}] \wedge t' \in [t_{\min}, t_n] \wedge r < x'_{\min} \wedge t < t')$;
2. When $t \leq t_1$ and $t' \geq t_n$, the robustness value is given by $\min\{w_x[t_1], w_x[t_n]\}$. More formally, when $t_1 > \inf J_1$ and $t_n < \sup J_n$ (either both will be true or, if $x'_{\min} = w_x[\inf J_1] = w_x[\sup J_n]$, both will be false), we add to $\mathcal{S}_{x \geq 0}$ two polyhedra: $(t \in [\inf J_1, t_1] \wedge t < t' \wedge a_1 t + b_1 \leq a_2 t' + b_2 \wedge r \leq a_1 t + b_1)$ and $(t' \in [t_n, \sup J_n] \wedge t < t' \wedge a_2 t' + b_2 \leq a_1 t + b_1 \wedge r \leq a_n t' + b_n)$. This also accounts for the cases where $t \leq t_1$, but $t' < t_n$ (robustness is $w_x[t]$) and where $t > t_1$ and $t' \geq t_n$ (robustness is $w_x[t']$).
3. If $t_{\min} > t_1$, we apply the procedure recursively to the interval $J = [t_1, t_{\min}]$ (there may be a degenerate case where $t_{\min} = t_1 = \sup J_1$);
4. If $t_{\min} < t_n$, we apply the procedure recursively to the interval $J = [t_{\min}, t_n]$ (there may be a degenerate case where $t_{\min} = t_n = \inf J_n$).

One can show that this procedure produces a number of polyhedra linear in the number of segments of the signal. The total number of recursive calls of the second and third kind is bounded by the number of switching points, since every call consumes the interior point of a wedge or a point of local minimum and this point cannot appear in either role in the other recursive calls. Additionally, we make a number of recursive calls of the first kind bounded by the number of segments. Also, observe that in the resulting polyhedra, r is always unbounded from below.

Other Expressions. Robustness support can be characterized by induction on the expression structure using basic set operations. The rules below can be derived from this inductive characterization.

Empty word. For the expression ϵ, one can show that $\mathcal{R}(\epsilon, w) = \{(t, t', r) \mid t = t'\}$, which is represented using the singleton *zone*: $\mathcal{S}_\epsilon = \{(t = t')\}$.

Falsehood. In this case, robustness support is empty, thus $\mathcal{S}_\varnothing = \emptyset$.

Disjunction For the disjunction $\varphi \vee \psi$, we have $\mathcal{R}(\varphi \vee \psi, w) = \mathcal{R}(\varphi, w) \cup \mathcal{R}(\psi, w)$, thus we take the union of the sets of polyhedra, $\mathcal{S}_{\varphi \vee \psi} = \mathcal{S}_\varphi \cup \mathcal{S}_\psi$.

Conjunction For the conjunction $\varphi \wedge \psi$, $\mathcal{R}(\varphi \wedge \psi, w) = \mathcal{R}(\varphi, w) \cap \mathcal{R}(\psi, w)$, thus we take the pairwise intersection: $\mathcal{S}_{\varphi \wedge \psi} = \bigcup_{\substack{P_\varphi \in \mathcal{S}_\varphi \\ P_\psi \in \mathcal{S}_\psi}} \{P_\varphi \cap P_\psi\}$

Concatenation. Notice that we identify polyhedra (or zones) with conjunctions of constraints. Thus, for two polyhedra P and Q, the formula $P \wedge Q$ denotes the polyhedron that is the intersection of P and Q. For a polyhedron P with 4 dimensions t, t', t'' and r, the formula $\exists t''.\ P[t, t', t'', r]$ denotes the polyhedron that is the projection of P on t, t', r. For concatenation, one can show that $\mathcal{R}(\varphi \cdot \psi, w) = \{(t, t', r) \mid \exists t''.\ (t, t'', r) \in$

$\mathcal{R}(\varphi, w) \wedge (t'', t', r) \in \mathcal{R}(\psi, w)\}$. Thus, $\mathcal{S}_{\varphi \cdot \psi}$ is the set of polyhedra $\{\exists t''.\ P_\varphi[t, t'', r] \wedge P_\psi[t'', t', r] \mid P_\varphi \in \mathcal{S}_\varphi,\ P_\psi \in \mathcal{S}_\psi\}$.

Kleene star. From Theorem 3 it follows that for every signal w and expression φ, there exists $k \geq 0$, s.t. (i) $\mathcal{R}(\varphi^k, w) \supseteq \mathcal{R}(\varphi^{k+1}, w)$; and (ii) $\mathcal{R}(\varphi^*, w) = \bigcup_{i=0}^{k} \mathcal{R}(\varphi^i, w)$. By definition of the Kleene star, $\mathcal{R}(\varphi^*, w) = \bigcup_{k \geq 0} \mathcal{R}(\varphi^k, w)$, hence $\mathcal{S}_{\varphi^*} = \bigcup_{i=0}^{k} \mathcal{S}_{\varphi^i}$ where k is the smallest index, s.t. $\mathcal{R}(\varphi^k, w) \supseteq \mathcal{R}(\varphi^{k+1}, w)$. A sufficient stopping condition thus is to check whether the set of polyhedra $\mathcal{S}_{\varphi^k}$ *geometrically covers* the set of polyhedra $\mathcal{S}_{\varphi^{k+1}}$ (it can be implemented, e.g., by checking intersection with the complement). In [32], it was shown that for the pattern matching problem, a sufficient stopping condition can be formulated using pairwise inclusion rather than geometric coverage. Whether this is also true in the case of *robust* pattern matching relative to arbitrary polyhedra, is yet to be shown.

Duration Restriction. For the duration restriction $\langle \varphi \rangle_I$, one can show $\mathcal{R}(\langle \varphi \rangle_I, w) = \mathcal{R}(\varphi, w) \cap \{(t, t', r) \mid t' - t \in I\}$. Thus, we restrict every element of $\mathcal{S}_\varphi$ as follows: $\mathcal{S}_{\langle \varphi \rangle_I} = \{P \wedge (t' - t \in I) \mid P \in \mathcal{S}_\varphi\}$.

Proof (of Theorem 4). We can now prove Theorem 4. For a piecewise-constant signal w and an expression φ, $\mathcal{S}_\varphi$ is a set of *zones* over t, t', r, s.t. every constraint on r is of the form $r < c$. This holds for the base cases and is preserved by intersection and projection operations performed for the inductive cases. When a zone Z, bounds r from above by c (possibly ∞), the pointwise supremum function $(t, t') \mapsto \sup\{r \mid (t, t', r) \in Z\}$ is piecewise-constant. The robustness map $(t, t') \mapsto \rho(\varphi, w, [t, t'))$ is a pointwise maximum of finitely many piecewise-constant functions, and is piecewise-constant. Similarly, for a continuous piecewise-linear signal, $\mathcal{S}_\varphi$ is a set of polyhedra, where r is unbounded from below. For every such polyhedron P, the function $(t, t') \mapsto \sup\{r \mid (t, t', r) \in P\}$ is piecewise-linear. The robustness map is a pointwise maximum of finitely many piecewise-linear functions and is piecewise-linear. □

Possible Optimizations A practical issue of computing robustness support in this bottom-up way is that we have to compute and store the robustness value for every segment of the signal and every sub-expression, regardless of how small this value is and whether it will be used when matching the higher-level expressions. One workaround is to approximate robustness by replacing the values that are below some threshold with $-\infty$ (discarding the corresponding polyhedra). Another optimization is to rewrite the original expression and propagate time restriction operations to the sub-expressions. This will allow to earlier discard the polyhedra that would anyway be discarded by the time restriction later in the computation.

6 Experiments

In this section, we evaluate our matching algorithms on a problem of finding ringing patterns in a signal. Ringing is a damped oscillation of an output of a system as a response to a sudden change in the input. In Figure 6 (left), we give an example of a ringing behavior of a linear system with respect to a square wave input. We define

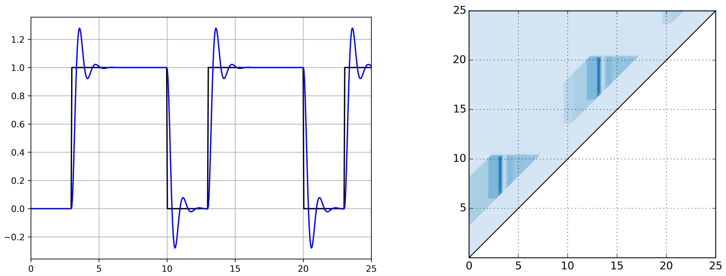

Fig. 6. A ringing signal w (left) and its robustness map (right).

ringing using the following expression:

$$\langle x \leq 0.2 \rangle_{\leq 0.05} \cdot \langle 0.1 \leq x \leq 0.9 \rangle_{\leq 0.05} \cdot \langle 0.7 \leq x \leq 1.3 \rangle_{[0.3,\ 1]} \cdot \langle 0.9 \leq x \leq 1.1 \rangle_{[3,\ 6]}$$

This is a concatenation of constraints that describe (from left to right) low value, rising edge, ringing, and stable high value periods, using thresholds on the value and duration. We discretize the input signal and feed it to our matching tool Montre [31], which was extended to support robust matching of piecewise-constant signals. In Figure 6 (right), we show the robustness map produced by Montre, where darker colors correspond to time segments with higher robustness values. In particular, thin dark bands correspond to the segments that start on the rising edge of the signal and end at high signal value. These segments also satisfy the expression qualitatively. The surrounding lighter areas correspond to the segments that do not satisfy the expression qualitatively, but are close to satisfaction. These are the segments that start before the rising edge of the signal, or during or after the second ringing oscillation.

To evaluate the practical complexity of robust matching, we generate longer input signals, consisting of multiple square waves with ringing. In Table 1, we give execution times and numbers of output zones for different lengths of the input (measured in the number of discrete samples). We observe that for this example the runtime of the algorithm is linear in the length of the input, which is expected when the duration constraints in the expression are much shorter than the input itself. Additionally (not shown in Table 1) we measured the overhead of performing matching on piecewise-linear interpolation of a signal compared to piecewise-constant, which is due to using polyhedra instead of zones. To represent and manipulate polyhedra, we use Parma Polyhedra Library (PPL) [7]. Our experiments with PPL suggest that individual operations on polyhedra (corresponding to SRE operators) are 30 to 40 times slower than those on

Table 1. Experimental Results

Input length	10K	20K	40K
Execution Time (sec)	3.88	7.80	15.5
Number of output zones	156K	315K	631K

zones. The implementation of zones is optimized for the particular form of constraints, while for polyhedra, PPL implements the double-description method [8], where a polyhedron is represented by a system of constraints and a system of generators.

7 Conclusion

This work can be seen as part of the trend of extending formal language theory and its related formalisms towards the quantitative; somewhat in the spirit of [29,30] (for metric time) and [18,19]. Our first contribution is to introduce a quantitative semantics for SRE which indicates how robustly a signal matches or does not match a given expression. This semantics, which is a safe approximation of the uniform norm distance between the signal and the expression, can also be applied in discrete time to characterize the robustness of the membership of a sequence of numbers in a regular language over a numerical alphabet. We then define the problem of robust pattern matching, i.e., determining the robustness of every segment of the input signal. This problem arises naturally when computing the robustness of a signal for an expression containing concatenation. Moreover, this additional information can be very useful in novel applications of such specification formalisms, for example [21,11,1]. Unlike classical verification, where we want to verify properties of the whole behavior, in monitoring of real systems or in data mining, we would like to detect the occurrence of patterns at various parts of the signal. We developed algorithms to solve this robust matching problem for two classes of signals: piecewise-constant and continuous piecewise-linear, which are both common in digital signal processing based on sampled signals. We observe that robust pattern matching can be seen as constructing a 3-dimensional surface, and show that this surface is piecewise-linear for piecewise-linear signals. Practically, we represent the volume under this surface as a set of zones or convex polyhedra.

Future Work. We consider the following directions for future work. First, we observe that our definition of robustness represents purely *spatial* distance and thus does not address *time* robustness (see discussion in [15]). This issue is best demonstrated with the duration operator. Consider an expression $\langle \varphi \rangle_{[2,3]}$ and a signal that matches φ but with duration $2 - \delta$, for a small δ. With our current semantics, its robustness is $-\infty$, regardless of how small δ is. An alternative more continuous semantics of the time restriction operator will bring our semantics closer to other (non-pointwise) metrics that allow stretching and shrinking of behaviors [12,20].

Also, as mentioned in the end of Section 5, unlike the very efficient qualitative matching [32], our quantitative matching algorithm is costly. One reason is the use of arbitrary convex polyhedra for piecewise-linear signals. Another is the fact that we need to cover the whole triangle $0 \leq t \leq t' \leq |w|$ by polyhedra or zones, while for the qualitative semantics, the match set is very sparse. In addition to the ad-hoc optimizations mentioned in the end of Section 5, we foresee two more rigorous ways to address this problem. First, we can use an approximate semantics that quantizes the robustness values into a finite set of ranges. Second, if we are only interested in the robustness with respect to the whole signal, we may be able to replace the current bottom-up algorithm by a more sophisticated top-down scheme that will compute robustness only with respect to a subset of the sub-expressions and time segments.

References

1. Abbas, H., Rodionova, A., Bartocci, E., Smolka, S.A., Grosu, R.: Regular expressions for irregular rhythms. arXiv preprint arXiv:1612.07770 (2016)
2. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science 126(2), 183–235 (1994)
3. Alur, R., Feder, T., Henzinger, T.A.: The benefits of relaxing punctuality. J. ACM 43(1), 116–146 (1996)
4. Alur, R., Fisman, D., Raghothaman, M.: regular programming for quantitative properties of data streams. In: European Symposium on Programming Languages and Systems. pp. 15–40. Springer Berlin Heidelberg (2016)
5. Asarin, E., Caspi, P., Maler, O.: Timed regular expressions. J. ACM 49(2), 172–206 (2002)
6. Asarin, E., Caspi, P., Maler, O.: A Kleene theorem for timed automata. In: Logic in Computer Science (LICS). pp. 160–171 (1997)
7. Bagnara, R., Hill, P.M., Zaffanella, E.: The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. Science of Computer Programming 72(1–2), 3–21 (2008)
8. Bagnara, R., Hill, P.M., Zaffanella, E.: Not necessarily closed convex polyhedra and the double description method. Formal Asp. Comput. 17(2), 222–257 (2005)
9. Basin, D., Klaedtke, F., Müller, S., Zălinescu, E.: Monitoring metric first-order temporal properties. Journal of the ACM (JACM) 62(2), 15 (2015)
10. D'Antoni, L., Veanes, M.: Minimization of symbolic automata. In: POPL. pp. 541–554. ACM (2014)
11. Deshmukh, J.V., Jin, X., Seshia, S., et al.: Learning auditable features from signals using unsupervised temporal projection (2017)
12. Deshmukh, J.V., Majumdar, R., Prabhu, V.S.: Quantifying conformance using the skorokhod metric. In: International Conference on Computer Aided Verification. pp. 234–250. Springer (2015)
13. Dill, D.L.: Timing assumptions and verification of finite-state concurrent systems. In: Sifakis, J. (ed.) Automatic Verification Methods for Finite State Systems. LNCS, vol. 407, pp. 197–212. Springer (1989)
14. Donzé, A.: Breach, a toolbox for verification and parameter synthesis of hybrid systems. In: CAV. pp. 167–170 (2010)
15. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: FORMATS 2010. LNCS, vol. 6246, pp. 92–106. Springer (2010)
16. Donzé, A., Ferrère, T., Maler, O.: Efficient robust monitoring for STL. In: Computer Aided Verification (CAV). pp. 264–279 (2013)
17. Fainekos, G.E., Pappas, G.J.: Robustness of temporal logic specifications. In: Proceedings of FATES/RV. LNCS, vol. 4262, pp. 178–192. Springer (2006)
18. Henzinger, T.A.: Quantitative reactive modeling and verification. Computer Science-Research and Development 28(4), 331–344 (2013)
19. Henzinger, T.A., Otop, J.: From model checking to model measuring. In: Conference on Concurrency Theory (CONCUR). pp. 273–287 (2013)
20. Jakšić, S., Bartocci, E., Grosu, R., Ničković, D.: Quantitative monitoring of STL with edit distance. In: International Conference on Runtime Verification. pp. 201–218. Springer International Publishing (2016)
21. Jin, X., Donzé, A., Deshmukh, J.V., Seshia, S.A.: Mining requirements from closed-loop control models. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 34(11), 1704–1717 (2015)

22. Koymans, R.: Specifying real-time properties with metric temporal logic. Real-Time Syst. 2(4), 255–299 (1990)
23. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: FORMATS/FTRTFT. pp. 152–166 (2004)
24. Maler, O., Nickovic, D., Pnueli, A.: Checking temporal properties of discrete, timed and continuous behaviors. In: Pillars of Computer Science. pp. 475–505 (2008)
25. Mens, I.E., Maler, O.: Learning regular languages over large ordered alphabets. Logical Methods in Computer Science (LMCS) 11(3)
26. Nickovic, D., Maler, O.: AMT: A property-based monitoring tool for analog systems. In: FORMATS. pp. 304–319 (2007)
27. Pnueli, A.: The temporal logic of programs. In: Proc. 18th Annual Symposium on Foundations of Computer Science (FOCS). pp. 46–57 (1977)
28. Rizk, A., Batt, G., Fages, F., Soliman, S.: On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology. In: CMSB. pp. 251–268 (2008)
29. Trakhtenbrot, B.A.: Origins and metamorphoses of the trinity: Logic, nets, automata. In: LICS. pp. 506–507 (1995)
30. Trakhtenbrot, B.A.: Understanding basic automata theory in the continuous time setting. Fundam. Inform. 62(1), 69–121 (2004)
31. Ulus, D.: Montre: A tool for monitoring timed regular expressions. CoRR abs/1605.05963 (2016)
32. Ulus, D., Ferrère, T., Asarin, E., Maler, O.: Timed pattern matching. In: Formal Modeling and Analysis of Timed Systems (FORMATS). pp. 222–236 (2014)
33. Veanes, M., Hooimeijer, P., Livshits, B., Molnar, D., Björner, N.: Symbolic finite state transducers: algorithms and applications. In: POPL. pp. 137–150. ACM (2012)