Monitoring First-Order Properties of Real-Valued Signals

Alexey Bakhirkin[1], Thomas Ferrère[2], Thomas A. Henzinger[2], and Dejan Nickovic[3]

[1] VERIMAG – University of Grenoble, CNRS, and INPG
[2] IST Austria
[3] AIT Austrian Institute of Technology

Property-based monitoring is a pragmatic, yet rigorous approach to reason about complex systems, combining formal specifications with the analysis of individual system behaviors. Signal temporal logic (STL) [3] is a specification formalism for expressing real-time temporal properties of real-valued signals. The *bounded stabilization* requirement is a typical example of a temporal specification than can be expressed in STL. Given a Boolean signal b and a real-valued signal f, the bounded stabilization requirement is formulated as follows: "Whenever the control signal b is on its rising edge, the absolute value of f must go inside the interval $[4.5, 5.5]$ within 10 time units and continuously remain within that same interval for at least 8 time units". This informal requirement, illustrated in Figure 1-(a), is expressed as the following STL specification.

$$\varphi \equiv \Box(\uparrow b \rightarrow \Diamond_{[0,10]} \Box_{[0,8]} (|f - 5| \leq 0.5)$$

The expressiveness of STL has nevertheless some limitations. For instance, the bounded stabilization property requires apriori knowledge of thresholds and timing bounds. In real-life applications, these bounds may not be known or may even change dynamically. A more general formulation of the bounded stabilization property requires the signal f to stabilize around *some* value of r, which can vary during the execution of the system as shown in Figure 1-(b). This is a common property that cannot be expressed in STL nor by any other logic-based formalism for monitoring real-valued signals in the litterature. The general bounded stabilization requirement could be formulated in STL extended with quantification over the threshold value as follows.

$$\psi \equiv \Box(\uparrow b \rightarrow \exists r : \Diamond_{[0,10]} \Box_{[0,8]} (|f - r| \leq 0.5)$$

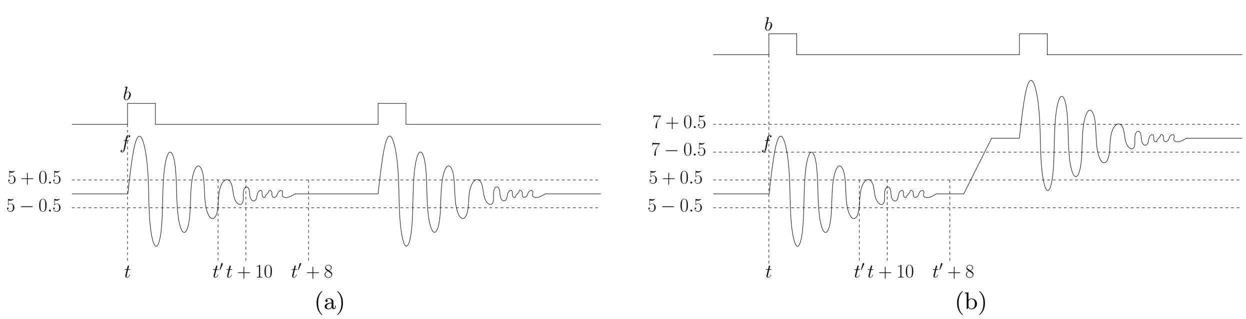

Fig. 1. Bounded stabilization (a) with fixed threshold $r = 5$; (b) with variable threshold r.

This form of quantification is slight generalization of the logic of [2], which provides STL with a restricted form of quantification called *value-freezing*. By convention, we use r for value variables and s for time variables with t the special free variable standing for absolute time. Remark that when using *time* parameters, temporal logic operators become unnecessary because the quantification over time implicit in temporal operators is then made explicit in syntax. In particular, by using first-order quantification, the modality $\Diamond_{[s,s]}$, where s is a free variable, can express all other forms of temporal operators. Motivated by the lack of a clean specification language that is sufficiently expressive to capture rich temporal properties, we propose *signal first-order logic* (SFO) as a powerful declarative formalism for expressing real-valued signal requirements. SFO consist in first-order formulas based on linear arithmetic predicates over real variables and uninterpreted function symbols, standing for real-valued signals.

The bounded stabilization property is expressed in SFO as follows.

$$\psi' \equiv \uparrow b \rightarrow \exists r : \exists s_1 \in [0, 10] : \forall s_2 \in [0, 8] : |f(t + s_1 + s_2) - r| \leq 0.5$$

where $\uparrow b \equiv b(t) = 1 \land \exists s_1 \in (0,1) : \forall s_2 \in (0, s_1) : b(t - s_2) = 0$. We can also express the control property that whenever f_1 is stable then f_2 becomes stable around the same value, as follows.

$$\gamma \equiv \forall r : (\forall s_1 \in [0, 10] : |f_1(t + s_1) - r| \leq 1) \rightarrow (\forall s_2 \in [5, 10] : |f_2(t + s_2) - r| \leq 2)$$

In more details, formula γ requires that if f_1 stays within 1.0 of some value r for 10 time units then f_2 stays within 2.0 of r within 5 time units.

Since SFO easily encode STL and other expressive logics, automated reasonning on SFO specification is not possible in general.

Theorem 1. *The satisfiability of SFO is undecidable.*

However we are only interested in using SFO for offline monitoring, defined as follows.

Definition 1. *The* satisfaction signal *of formula φ relative to trace w is the Boolean signal denoted w_φ such that $w_\varphi(t) = 1$ iff $(w, t) \models \varphi$ for all $t \in \mathbb{T}$. The* offline monitoring *problem is the task of computing, given a temporal formula φ and a signal w, the satisfaction signal of φ relative to w.*

We restrict our attention to piecewise-linear traces over a bounded time domain. The interpretation of every term and every formula can then be represented as a set of convex polyhedra. Our monitoring procedure of piecewise-linear signals for SFO over linear real arithmetic is based on this principle.

Functions Every function f is represented as a union of convex polyhedra $\mathcal{P}_f$ with two free variables: t_f denoting time and v_f denoting the value of f at the given time point.

Terms A term θ is represented as a union of convex polyhedra $\mathcal{P}_\theta$, with variables corresponding to free variables of the term and a fresh variable v_θ that corresponds to the value of the term.

- For $\theta \equiv \tau$ for some time linear expression τ, $\mathcal{P}_\theta = \{v_\theta = \tau\}$;
- For $\theta \equiv \rho$ for some space linear expression ρ, $\mathcal{P}_\theta = \{v_\theta = \rho\}$;
- For $\theta \equiv n$ for some constant n, $\mathcal{P}_\theta = \{v_\theta = n\}$;
- For $\theta \equiv f(\tau)$, $\mathcal{P}_\theta = \mathcal{P}_f[t_f \mapsto \tau, v_f \mapsto v_\theta]$;
- For $\theta \equiv \theta_1 \pm \theta_2$, $\mathcal{P}_\theta = \mathit{eliminate}(v_{\theta_1}, v_{\theta_2}, \{v_\theta = v_{\theta_1} \pm v_{\theta_2}\} \cap \mathcal{P}_{\theta_1} \cap \mathcal{P}_{\theta_2})$.

Formulas A formula φ is seen as a function from the values of its free variables to a Boolean value and thus can represented as a union of polyhedra $\mathcal{P}_\varphi$, with variables corresponding to free variables of the formula.

- For $\varphi \equiv \theta_1 < \theta_2$, $\mathcal{P}_\varphi = \mathit{eliminate}(v_{\theta_1}, v_{\theta_2}, \{v_{\theta_1} < v_{\theta_2}\} \cap \mathcal{P}_{\theta_1} \cap \mathcal{P}_{\theta_2})$;
- For $\varphi \equiv \neg\varphi'$, $\mathcal{P}_\varphi = \mathit{complement}(\mathcal{P}_{\varphi'})$;
- For $\varphi \equiv \varphi_1 \lor \varphi_2$, $\mathcal{P}_\varphi = \mathcal{P}_{\varphi_1} \cup \mathcal{P}_{\varphi_2}$;
- For $\varphi \equiv \exists r : \varphi'$, $\mathcal{P}_\varphi = \mathit{eliminate}(r, \mathcal{P}_{\varphi'})$;
- For $\varphi \equiv \exists t \in I : \varphi'$, $\mathcal{P}_\varphi = \mathit{eliminate}(t, \{t \in I\} \cap \mathcal{P}_{\varphi'})$.

Theorem 2. *The monitoring of SFO can be solved in time $2^{(m+n)^{2^{O(k+l)}}}$ over an n-long signal and m-long formula with k quantifiers and l function symbol occurrences, and in time $n2^{(m+j)^{2^{O(k+l)}}}$ when the formula is a bounded-response property and the signal has variability j.*

Most practical specifications belong to the bounded-time fragment. Specifications are typically concise, while traces are typically large, hence we believe that monitoring SFO is tractable in practice.

We implemented this algorithm using PPLite, an open-source polyhedra library based on PPL [1]. Preliminary experiments confirm the linear-time monitoring for the bounded-response fragment. In the future, we plan to release this implementation and compare it against monitoring tools for STL and variants.

References

1. Roberto Bagnara, Patricia M Hill, and Enea Zaffanella. 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.
2. Lubos Brim, P Dluhoš, D Šafránek, and Tomas Vejpustek. STL*: Extending signal temporal logic with signal-value freezing operator. *Information and Computation*, 236:52–67, 2014.
3. Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems (FORMATS/FTRTFT)*, pages 152–166, 2004.