Qualitative and Quantitative Monitoring of Spatio-Temporal Properties*

Laura Nenzi[1], Luca Bortolussi[2,3,4], Vincenzo Ciancia[4], Michele Loreti[5,1], and Mieke Massink[4]

[1] IMT, Lucca, Italy
[2] MOSI, Saarland University, Germany
[3] DMG, University of Trieste, Italy
[4] CNR-ISTI, Pisa, Italy
[5] DiSIA, University of Firenze, Italy
laura.nenzi@imtlucca.it

Abstract. We address the specification and verification of spatio-temporal behaviours of complex systems, introducing *Signal Spatio-Temporal Logic* (SSTL). This modal logic extends the *Signal Temporal Logic* with spatial operators capable of specifying topological properties in a discrete space. The latter is modelled as a weighted graph, and provided with a boolean and a quantitative semantics. Furthermore, we define efficient *monitoring algorithms* for both the boolean and the quantitative semantics. These are implemented in a Java tool available online. We illustrate the expressiveness of SSTL and the effectiveness of the monitoring procedures on the formation of patterns in a Turing reaction-diffusion system.
Keywords: Signal Spatio-Temporal Logic, Boolean Semantics, Quantitative Semantics, Monitoring Algorithms, Weighted Graphs, Turing Patterns.

1 Introduction

There is an increasing interest in the introduction of smart solutions in the world around us. A huge number of computational devices, located in space, is interacting in an open and changing environment, with humans and nature in the loop that form an intrinsic part of the system. Yet, science and technology are still struggling to tame the challenges underlying the design and control of such systems. In this paper, in particular, we focus on the challenge of spatially located systems, for which the spatial and temporal dimensions are strictly correlated and influence each other. This is the case in many Cyber-Physical Systems, like pacemaker devices controlling the rhythm of heart beat, and for many Collective Adaptive Systems, like the guidance of crowd movement in emergency situations or the improvement of the performance of bike sharing systems in smart cities.

Controlling and designing spatio-temporal behaviours requires proper formal tools to describe such properties, and to monitor and verify whether, and to which extent

* Work partially funded by the EU-FET project QUANTICOL (nr. 600708), by the German Research Council (DFG) as part of the Cluster of Excellence on Multimodal Computing and Interaction at Saarland University and the IT MIUR project CINA. We thank Diego Latella and Ezio Bartocci for the discussions and EB for sharing the code to generate traces of the example.

and how robustly, they are satisfied by a system. Formal methods play a central role, in terms of formal languages to specify spatio-temporal models and properties, and in terms of algorithms for the verification of such properties on such models and on monitored systems.

Related work. Logical specification and monitoring of temporal properties is a well-developed area. Here we mention Signal Temporal Logic (STL), an extension of Metric Interval Temporal Logic describing linear-time properties of real-valued signals. STL has monitoring routines both for its boolean and quantitative semantics, the latter measuring the satisfaction degree of a formula [7, 8, 14].

Much work has been done also in the area of spatial logic [1], yet focussing more on expressivity and decidability, often in continuous space. Less attention has been placed on more practical aspects, like model checking routines in discrete space. An exception is the work of some of the authors [4], in which the spatial logic SLCS (Spatial Logic for Closure Spaces) is proposed for a discrete and topological notion of space, based on closure spaces [10]. The spatial modal operator considered is the spatial until or surround, for which an efficient model checking routine is proposed. First applications of that work in the context of smart transportation can be found in [6]. Another spatial logic equipped with practical model checking algorithms, and with learning procedures, is that of [11, 12], in which spatial properties are expressed using ideas from image processing, namely quad trees. This allows them to capture very complex spatial structures, but at the price of a complex formulation of spatial properties, which are in practice only learned from some template image.

In this work, we will also focus on a notion of discrete space. The reason is that many applications, like bike sharing systems or metapopulation epidemic models [15], are naturally framed in a discrete spatial structure. Moreover, in many circumstances continuous space is abstracted as a grid or as a mesh. This is the case, for instance, in many numerical methods to simulate the spatio-temporal dynamics of Partial Differential Equations (PDE). Hence, this class of models is naturally dealt with by checking properties on such a discretisation.

The combination of spatial and temporal operators is even more challenging [1], and few works exist with a practical perspective. In [3], some of the authors proposed an extension of STL with a "somewhere" spatial modality, which can be arbitrarily nested with temporal operators, proposing a monitoring algorithm for both the boolean and the quantitative semantics. An extension of SLCS with temporal aspects can be found in [5] where the logic has been applied in the context of smart public transportation. In [13], instead, the authors merge the spatial logic of [12] within linear temporal logic, by considering atomic spatial properties. They also provide a qualitative and quantitative semantics, and apply it to smart grids and to the formation of patterns in a reaction diffusion model.

Contributions. In this work, we present a novel Spatio-Temporal logic, Signal Spatio-Temporal Logic (SSTL), that combines and extends the works of [3] and [4]. Our logic integrates the temporal modalities of STL, with the topological spatial surround and the somewhere modalities, imposing metric bounds on spatial distances and on temporal operators. We provide the logic with a qualitative and quantitative semantics, and we define monitoring algorithms for both of them. The major challenge is to monitor the surround operator, for which we propose two fixed point algorithms, one for the boolean and one for the quantitative semantics, discussing their correctness and computational cost. The monitoring algorithms have been implemented in Java, and applied and tested on a case study of pattern formation in a Turing reaction-diffusion system modelling a process of morphogenesis [13].

Paper structure. The paper is organised as follows: Section 2 introduces some background concepts on STL and on discrete topologies. Section 3 presents the syntax and the semantics of SSTL. Section 4 introduces the monitoring algorithms. Section 5 is devoted to the example of pattern formation, while conclusions are drawn in Section 6.

2 Background material

Weighted undirected graphs. We will consider discrete models of space that can be represented as a finite undirected graph. Edges of the graph are equipped with a positive weight, giving a metric structure to the space, in terms of shortest path distances. The weight will often represent the distance between two nodes. This is the case, for instance, when the graph is a discretization of continuous space. However, the notion of weight is more general, and may be used to encode different kinds of information. As an example, in a model where nodes are locations in the city and edges represent streets, the weight could represent the average travelling time, which can be different between two paths with the same physical length, but different levels of congestion or different number of traffic lights.

We represent a weighted undirected graph with a tuple $G = (L, E, w)$, where:

- L is the finite set of locations (nodes), $L \neq \emptyset$
- $E \subseteq L \times L$ is a symmetric relation, namely the set of connections (edges),
- $w : E \to \mathbb{R}_{>0}$ is the function that returns the cost/weight of each edge.

Furthermore, we denote by E^* the set containing all the pairs of connected locations, i.e. the transitive closure of E. We will also use an overloaded notation and extend w to the domain E^*, so that for arbitrary nodes x, y (not necessarily connected by an edge) we let $w(x, y)$ be the cost of the shortest path between two different locations. Finally, for all $\ell \in L$ and $w_1, w_2 > 0$, we let $L^{\ell}_{[w_1,w_2]}$ be the set of locations ℓ' such that $w_1 \leq w(\ell, \ell') \leq w_2$.

Closure spaces and the boundary of a set of nodes. In this work, we focus on graphs as an algorithmically tractable representation of space. However, *spatial* logics traditionally use more abstract structures, very often of a topological nature (see [1] for an exhaustive reference). We can frame a generalised notion of topology on graphs within the so called *Cech closure spaces*, a superclass of topological spaces allowing a clear formalisation of the semantics of the spatial surround operator on both topological and graph-like structures (see [4] and the references therein). What is really relevant for this work, due to the restriction to finite (weighted and undirected) graphs, is the notion of *external boundary* of a set of nodes A, i.e. the set of nodes directly connected with an element of A but not part of it.

Definition 1. *Given a subset of locations $A \subseteq L$, we define the boundary of A as:*

$$B^+(A) := \{\ell \in L | \ell \notin A \wedge \exists \ell' \in A \text{ s.t. } (\ell', \ell) \in E\}.$$

Signal Temporal Logic. *Signal Temporal Logic* (STL) is a linear dense time-bounded temporal logic that extends *Metric Interval Temporal Logic* (MITL) with a set of atomic propositions $\{\mu_1, ..., \mu_m\}$ that specify properties of real valued traces, therefore mapping real valued traces into boolean values.

Let $\mathbf{x} : \mathbb{T} \to \mathbb{D}$ a trace that describes an evolution of our system, where $\mathbb{T} = \mathbb{R}_{\geq 0}$ is the time set and $\mathbb{D} = \mathbb{D}_1 \times \cdots \times \mathbb{D}_n \subseteq \mathbb{R}^n$ is the domain of evaluation; then each $\mu_j : \mathbb{D} \to \mathbb{B}$ is of the form $\mu_j(x_1, \dots, x_n) \equiv (f_j(x_1, \dots, x_n) \geqslant 0)$, where $f_j : \mathbb{D} \to \mathbb{R}$ is a (possibly non-linear) real-valued function and $\mathbb{B} = \{true, false\}$ are boolean values. The projections $x_i : \mathbb{T} \to \mathbb{D}_i$ on the i^{th} coordinate/variable are called the *primary signals* and, for all j, the function $s_j : \mathbb{T} \to \mathbb{R}$ defined by point-wise application of f_j to the image of $\mathbf{x}$, namely $s_j(t) \coloneqq f_j(x_1(t), ..., x_n(t))$, is called a *secondary signal* [8].

The syntax of STL is given by

$$\varphi \coloneqq \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \, \mathcal{U}_{[t_1,t_2]} \, \varphi_2$$

where conjunction and negation are the standard boolean connectives, $[t_1, t_2]$ is a real positive dense intervals with $t_1 < t_2$, $\mathcal{U}_{[t_1,t_2]}$ is the *bounded until* operator and μ is an atomic proposition. The *eventually* operator $\mathcal{F}_{[t_1,t_2]}$ and the *always* operator $\mathcal{G}_{[t_1,t_2]}$ can be defined as usual: $\mathcal{F}_{[t_1,t_2]}\varphi \coloneqq \top\mathcal{U}_{[t_1,t_2]}\varphi$, $\mathcal{G}_{[t_1,t_2]}\varphi \coloneqq \neg\mathcal{F}_{[t_1,t_2]}\neg\varphi$.

3 SSTL: Signal Spatio-Temporal Logic

Signal Spatio-Temporal Logic (SSTL) is an extension of Signal Temporal Logic [7, 14] with two spatial modalities. The first one, the *bounded somewhere* operator $\diamondsuit_{[w1,w2]}$ is taken from [3], while the second one, the *bounded surround* operator $\mathcal{S}_{[w_1,w_2]}$, is inspired by the *Spatial Logic for Closure Spaces* [4]. In the following, we first introduce spatio-temporal signals, and then present the syntax and the boolean and quantitative semantics of SSTL.

Spatio-Temporal Signals. SSTL is interpreted on spatio-temporal, real-valued signals. Space is discrete and described by a weighted graph $G = (L, E, w)$, as in Section 2, while the time domain $\mathbb{T}$ will usually be the real-valued interval $[0, T]$, for some $T > 0$. A spatio-temporal trace is a function $\mathbf{x} : \mathbb{T} \times L \to \mathbb{D}$, where $\mathbb{D} \subseteq \mathbb{R}^n$ is the domain of the trace. As for temporal traces, we write $\mathbf{x}(t, \ell) = (x_1(t, \ell), \cdots, x_n(t, \ell)) \in \mathbb{D}$, where each $x_i : \mathbb{T} \times L \to \mathbb{D}_i$, for $i = 1, ..., n$, is the projection on the i^{th} coordinate/variable.

Spatio-temporal traces can be obtained by simulating a stochastic model or a deterministic model, i.e. specified by a set of differential equations. In previous work [3], some of the authors discussed the framework of patch-based population models, which generalise population models and are a natural setting from which both stochastic and deterministic spatio-temporal traces of the considered type emerge. An alternative source of traces are measurements of real systems. For the purpose of this work, it is irrelevant which is the source of traces, as we are interested in their off-line monitoring.

Spatio-temporal traces are then converted into spatio-temporal boolean or quantitative signals. Similarly to the case of STL, each *atomic predicate* μ_j is of the form $\mu_j(x_1, \dots, x_n) \equiv (f_j(x_1, \dots, x_n) \geq 0)$, for $f_j : \mathbb{D} \to \mathbb{R}$. Each atomic proposition gives rise to a spatio-temporal signal. In the boolean case, one may define function

$s_j : \mathbb{T} \times L \to \mathbb{B}$; given a trace $\mathbf{x}$, this gives rise to the boolean signal $s_j(t,\ell) = \mu_j(\mathbf{x}(t,\ell))$ by point-wise lifting. Similarly, a quantitative signal is obtained as the real-valued function $s_j : \mathbb{T} \times L \to \mathbb{R}$, with $s_j(t,\ell) = f_j(\mathbf{x}(t,\ell))$.

When the space L is finite, as in our case, we can represent a spatio-temporal signal as a finite collection of temporal signals. More specifically, the signal $s(t,\ell)$ can be equivalently represented by the collection $\{s_\ell(t) \mid \ell \in L\}$. We will stick mostly to this second notation in the following, as it simplifies the presentation.

Syntax. The syntax of SSTL is given by

$$\varphi := \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \, \mathcal{U}_{[t_1,t_2]} \, \varphi_2 \mid \; \diamondsuit_{[w_1,w_2]} \, \varphi \mid \varphi_1 \, \mathcal{S}_{[w_1,w_2]} \varphi_2.$$

Atomic predicates, boolean operators, and the until operator $\mathcal{U}_{[t_1,t_2]}$ are those of STL. The spatial operators are the *somewhere* operator, $\diamondsuit_{[w_1,w_2]}$, and the *bounded surround* operator $\mathcal{S}_{[w_1,w_2]}$, where $[w_1,w_2]$ is a closed real interval with $w_1 < w_2$. The spatial somewhere operator $\diamondsuit_{[w_1,w_2]}\varphi$ requires φ to hold in a location reachable from the current one with a total cost greater than or equal to w_1 and less than or equal to w_2. The surround formula $\varphi_1 \, \mathcal{S}_{[w_1,w_2]}\varphi_2$ is true in a location ℓ, for the trace $\mathbf{x}$, when ℓ belongs to a set of locations A satisfying φ_1, such that its external boundary $B^+(A)$ (i.e., all the nearest neighbours of locations in A) contain only locations satisfying φ_2. Furthermore, locations in $B^+(A)$ must be reached from ℓ by a shortest path of cost between w_1 and w_2. Hence, the surround operator expresses the topological notion of being surrounded by a φ_2-region, with additional metric contraints. We can also derive the *everywhere* operator $\boxdot_{[w_1,w_2]}\varphi := \neg \diamondsuit_{[w_1,w_2]} \neg\varphi$ requiring φ to hold in all the locations reachable from the current one with a total cost between w_1 and w_2.

Semantics. We now define the boolean and the quantitative semantics for SSTL. The boolean semantic, as customary, returns true/false depending on whether the observed trace satisfies the SSTL specification.

Definition 2 (Boolean semantics). *The boolean satisfaction relation for an SSTL formula φ over a spatio-temporal trace $\mathbf{x}$ is given by:*

$$\begin{array}{lcl}
(\mathbf{x},t,\ell) \models \mu & \Leftrightarrow & \mu(\mathbf{x}(t,\ell)) = 1 \\
(\mathbf{x},t,\ell) \models \neg\varphi & \Leftrightarrow & (\mathbf{x},t,\ell) \not\models \varphi \\
(\mathbf{x},t,\ell) \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow & (\mathbf{x},t,\ell) \models \varphi_1 \textit{ and } (\mathbf{x},t,\ell) \models \varphi_2 \\
(\mathbf{x},t,\ell) \models \varphi_1 \, \mathcal{U}_{[t_1,t_2]}\varphi_2 & \Leftrightarrow & \exists t' \in [t+t_1, t+t_2] \textit{ s.t. } (\mathbf{x},t',\ell) \models \varphi_2 \\
 & & \quad \textit{and } \forall t'' \in [t,t'], (\mathbf{x},t'',\ell) \models \varphi_1 \\
(\mathbf{x},t,\ell) \models \diamondsuit_{[w_1,w_2]}\varphi & \Leftrightarrow & \exists \ell' \in L \textit{ s.t. } (\ell',\ell) \in E^*, \\
 & & \quad w_1 \leqslant w(\ell',\ell) \leqslant w_2 \textit{ and } (\mathbf{x},t,\ell') \models \varphi \\
(\mathbf{x},t,\ell) \models \varphi_1 \, \mathcal{S}_{[w_1,w_2]}\varphi_2 & \Leftrightarrow & \exists A \subseteq L^{\ell}_{[0,w_2]} \textit{ s.t. } \ell \in A \textit{ and } \forall \ell' \in A, (\mathbf{x},t,\ell') \models \varphi_1 \\
 & & \quad \textit{and } B^+(A) \subseteq L^{\ell}_{[w_1,w_2]} \textit{ and } \forall \ell'' \in B^+(A), (x,t,\ell'') \models \varphi_2.
\end{array}$$

A trace $\mathbf{x}$ satisfies φ in location ℓ, denoted by $(\mathbf{x},\ell) \models \varphi$, if and only if $(\mathbf{x},0,\ell) \models \varphi$.

The quantitative semantics returns a real value that can be interpreted as a measure of the strength with which the specification is satisfied or falsified by an observed trajectory. More specifically, the sign of such a satisfaction score is related to the truth of the formula (positive stands for true), while the absolute value of the score is a measure of the robustness of the satisfaction or dissatisfaction. This definition of quantitative measure is based on [7, 8], and it is a reformulation of the robustness degree of [9].

Definition 3 (SSTL Quantitative Semantics). *The quantitative satisfaction function* $\rho(\varphi, \mathbf{x}, t)$ *for an SSTL formula* φ *over a spatio-temporal trace* $\mathbf{x}$ *is given by:*

$$
\begin{aligned}
\rho(\mu, \mathbf{x}, t, \ell) &= f(\mathbf{x}(t, \ell)) \quad \textit{where } \mu \equiv (f \geq 0) \\
\rho(\neg\varphi, \mathbf{x}, t, \ell) &= -\rho(\varphi, \mathbf{x}, t, \ell) \\
\rho(\varphi_1 \wedge \varphi_2, \mathbf{x}, t, \ell) &= \min(\rho(\varphi_1, \mathbf{x}, t, \ell), \rho(\varphi_2, \mathbf{x}, t, \ell)) \\
\rho(\varphi_1 \, \mathcal{U}_{[t_1,t_2]} \varphi_2, \mathbf{x}, t, \ell) &= \sup_{t' \in t+[t_1,t_2]} (\min\{\rho(\varphi_2, \mathbf{x}, t', \ell), \inf_{t'' \in [t,t']} (\rho(\varphi_1, \mathbf{x}, t'', \ell))\} \\
\rho(\diamondsuit_{[w_1,w_2]} \varphi, \mathbf{x}, t, \ell) &= \max\{\rho(\varphi, \mathbf{x}, t, \ell') \mid \ell' \in L, (\ell', \ell) \in E^* \\
&\quad \textit{and } w_1 \leqslant w(\ell', \ell) \leqslant w_2\} \\
\rho(\varphi_1 \, \mathcal{S}_{[w_1,w_2]} \varphi_2, \mathbf{x}, t, \ell) &= \max_{A \subseteq L^{\ell}_{[0,w_2]}, \ell \in A, B^+(A) \subseteq L^{\ell}_{[w_1,w_2]}} (\min(\min_{\ell' \in A} \rho(\varphi_1, \mathbf{x}, t, \ell'), \\
&\quad \min_{\ell'' \in B^+(A)} \rho(\varphi_2, \mathbf{x}, t, \ell''))).
\end{aligned}
$$

The satisfaction score has some fundamental properties: if $\rho(\varphi, \mathbf{x}, t) > 0$, then $(\mathbf{x}, t, \ell) \models \varphi$, and similarly if $\rho(\varphi, \mathbf{x}, t) < 0$, then $(\mathbf{x}, t, \ell) \not\models \varphi$. The absolute value $|\rho(\varphi, \mathbf{x}, t)|$, instead, gives a measure of the strength of the truth value. The definition for the surround operator is essentially obtained from the boolean semantics by replacing conjunctions and universal quantifications with the minimum and disjunctions and existential quantifications with the maximum, as done in [7, 8] for STL.

4 Monitoring Algorithms

In this section we present the monitoring algorithms to check the validity of a formula φ on a trace $\mathbf{x}(t, \ell)$. The monitoring procedure, which is similar to the ones for STL [8, 14], works inductively bottom-up on the parse tree of the formula. In the case of the boolean semantics, for each subformula ψ, it constructs a signal s_ψ s.t. $s_\psi(\ell, t) = 1$ iff the subformula is true in position ℓ at time t. In the case of the quantitative semantics, for each subformula ψ, the signal s_ψ corresponds to the value of the quantitative satisfaction function ρ, for any time t and location ℓ. In this paper, we discuss the algorithms to check the bounded surround operator. The procedures for the boolean and temporal operators are those of STL [7, 8, 14], while the methods for the somewhere spatial modality have been previously discussed in [3], and are a simple extension of the procedure for the boolean operators. The treatment of the bounded surround modality $\psi = \varphi_1 \mathcal{S}_{[w_1,w_2]} \varphi_2$, instead, deviates substantially from these procedures. In the following, we will present two recursive algorithms to compute the boolean and the quantitative satisfaction, taking inspiration from [4] and assuming the knowledge of the boolean/quantitative signals of φ_1 and φ_2.

Preliminary notions on boolean signals. Before describing the algorithm 1, we need to introduce the definition of *minimal interval covering* $\mathcal{I}_{s_1,\ldots,s_n}$ *consistent with a set of temporal signals* $s_1, \ldots s_n$, see also [14].

Definition 4. *Given an interval* I*, and a set of temporal signals* $s_1, \ldots s_n$ *with* $s_i : I \to \mathbb{B}$*, the* **minimal interval covering** $\mathcal{I}_{s_1,\ldots,s_n}$ **of** I **consistent with the set of signals** $s_1, \ldots, s_n$ *is the shortest finite sequence of left-closed right-open intervals* $I_1, ..., I_h$ *such that* $\bigcup_j I_j = I$*,* $I_i \bigcap I_j = \emptyset$*,* $\forall i \neq j$*, and for* $k \in \{1, \ldots, n\}$*,* $s_k(t) = s_k(t')$ *for all* t*,* t' *belonging to the same interval. The* **positive minimal interval covering** *of* s *is* $\mathcal{I}^+_s = \{I \in \mathcal{I}_s \big| \forall t \in I : s(t) = 1\}$.

Monitoring the Boolean semantics of the bounded surround. Algorithm 1 presents the procedure to monitor the boolean semantics of a surround formula $\psi = \varphi_1 \mathcal{S}_{[w_1,w_2]} \varphi_2$ in a single location $\hat{\ell}$, returning the boolean signal $s_{\psi,\hat{\ell}}$ of ψ at location $\hat{\ell}$. The algorithm first computes the set of locations $L^{\hat{\ell}}_{[0,w_2]}$ that are at distance w_2 or less from $\hat{\ell}$, and then, recursively, the boolean signals $s_{\varphi_1,\ell}$ and $s_{\varphi_2,\ell}$, for $\ell \in L^{\hat{\ell}}_{[0,w_2]}$. These signals provide the satisfaction of the sub-formula φ_j at each point in time, and for each location of interest. Then, a minimal interval covering consistent to all the signals $s_{\varphi_1,\ell}$ and $s_{\varphi_2,\ell}$ is computed, and to each such interval, a core procedure similar to that of [4] is applied. More specifically, we first compute the set of locations T in which both φ_1 and φ_2 are false, and that are in the external boundary of the locations that satisfy φ_1 (V) or φ_2 (Q). The locations in T are "bad" locations, that cannot be part of the external boundary of the set A of φ_1-locations which has to be surrounded only by φ_2-locations. Hence, the main loop of the algorithm removes iteratively from V all those locations that have a neighbour in T (set N, line 13), constructing a new set T containing only those locations in N that do not satisfy φ_2, until a fixed point is reached. As each location can be added to T and be processed only once, the complexity of the algorithm is linear in the number of locations and linear in the size of the interval covering. Correctness can be proven in a similar way as in [4].

Piecewise constant approximation of quantitative signals. The quantitative semantics for STL is defined for arbitrary signals, but algorithms are provided for piecewise linear continuous ones [7, 8], considered as the interpolation of continuous functions. In this paper, we deviate from this interpretation, and consider instead a simpler interpolation based on piecewise constant signals. In particular, we discretise time with step $h > 0$, so that our signals in each location ℓ, $s_\ell : [0, T] \times L \to \mathbb{R}$, are represented by the finite set $\{s_\ell(0), s_\ell(h), m \ldots, s_\ell(mh)\}$, where $mh = T$. Then the piecewise constant approximation of $s_\ell(t)$ is the signal $\hat{s}_\ell(t) = s_\ell(kh)$ for $t \in [kh, (k+1)h)$. We further assume, without loss of generality[6], that all time bounds appearing in the temporal operators of a SSTL formula are multiples of h.

Under the assumption that secondary signals are Lipschitz continuous[7], and letting K be the maximum of their individual Lipschitz constants, we have that the following properties hold: (a) $s_\ell(kh) = \hat{s}_\ell(kh)$; and (b) $\|s_\ell(t) = \hat{s}_\ell(t)\| \leq Kh/2$, uniformly in t.

[6] Time bounds can be restricted to rational numbers, hence there always exists an $h > 0$ satisfying all assumptions.

[7] The assumption of Lipschitz continuity holds whenever the primary signal is the solution of an ODE with a locally Lipschitz vector field, as usually is the case.

Algorithm 1 Boolean monitoring for the surround operator

input $\hat{\ell}, \psi = \varphi_1 \mathcal{S}_{[w_1,w_2]} \varphi_2$
for all $\ell \in L^{\hat{\ell}}_{[0,w_2]}$ **do**
  compute recursively $s_{\varphi_1,\ell}, s_{\varphi_2,\ell}$,
end for
compute $\mathcal{I}_{s_{\psi,\hat{\ell}}}$ {the minimal interval covering consistent with $s_{\varphi_1,\ell}, s_{\varphi_2,\ell}, \quad \ell \in L^{\hat{\ell}}_{[0,w_2]}$}
for all $I_i \in \mathcal{I}_{s_{\psi,\hat{\ell}}}$ **do**
  $V = \{\ell \in L^{\hat{\ell}}_{[0,w_2]} | s_{\varphi_1,\ell}(I_i) = 1\}$
  $Q = \{\ell \in L^{\hat{\ell}}_{[w_1,w_2]} | s_{\varphi_2,\ell}(I_i) = 1\}$
  $T = B^+(Q \bigcup V)$
  while $T \neq \emptyset$ **do**
    $T' = \emptyset$
    for all $\ell \in T$ **do**
      $N = pre(\ell) \bigcap V = \{\ell' \in V | \ell E \ell'\}$
      $V = V \backslash N$
      $T' = T' \bigcup (N \backslash Q)$
    end for
    $T = T'$
  end while
  $s_{\psi,\hat{\ell}}(I_i) = \begin{cases} 1 & \text{if } \ell \in V, \\ 0 & \text{otherwise.} \end{cases}$
end for
merge adjacent positive interval I_i, i.e. I_i s.t. $s_{\psi,\hat{\ell}}(I_i) = 1$
return $s_{\psi,\hat{\ell}}$

Monitoring the quantitative semantics. We now turn to the monitoring algorithm for the quantitative semantics, assuming the input is a piecewise constant signal, where the time domain has been discretised with step h. Monitoring boolean operators is straightforward, we just need to apply the definition of the quantitative semantics pointwise in the discretisation. Monitoring the somewhere operator $\diamondsuit_{[w_1,w_2]}\varphi$ is also immediate: once the location $\hat{\ell}$ of interest is fixed, we can just turn into a disjunction of the signals $s_{\varphi,\ell}$ for each location $\ell \in L^{\hat{\ell}}_{[w_1,w_2]}$, see [3] for further details. The time bounded until operator, instead, can also be easily computed by replacing the min and max over dense real intervals in its definition by the corresponding min and max over the corresponding finite grid of time points. In this case, however, we can introduce an error due to the discrete approximation of the Lipschitz continuous signal in intermediate points, yet this error accumulates at a rate proportional to Kh, where K is the previously defined Lipschitz constant.

The only non-trivial monitoring algorithm is the one for the spatial surround operator, which will be discussed below. However, as the satisfaction score is computed at each time point of the discretisation and depends on the values of the signals at that time point only, this algorithm introduces no further error w.r.t. the time discretisation. Hence, we can globally bound the error introduced by the time discretisation:

Proposition 1. *Let the primary signal* $\mathbf{x}$ *be Lipschitz continuous, as the functions defining the atomic predicates. Let* K *be a Lipschitz constant for all secondary signals, and* h *be the discretisation step. Given a SSTL formula* φ*, let* $u(\varphi)$ *counts the number of temporal until operators in* φ*, and denote by* $\rho(\varphi, \mathbf{x})$ *its satisfaction score over the trace* $\mathbf{x}$ *and by* $\rho(\varphi, \hat{\mathbf{x}})$ *the satisfaction score over the discretised version* $\hat{\mathbf{x}}$ *of* $\mathbf{x}$ *with time step* h*. Then* $\|\rho(\varphi, \mathbf{x}) - \rho(\varphi, \hat{\mathbf{x}})\| \leq u(\varphi)Kh$*.*

Monitoring the quantitative semantics of the bounded surround. The quantitative monitoring procedure for the bounded surround operator is shown in Algorithm 2. Similarly to the boolean case, the algorithm for the surround formula $\psi = \varphi_1 \mathcal{S}_{[w_1,w_2]} \varphi_2$ takes as input a location $\hat{\ell}$ and returns the quantitative signal $s_{\psi,\hat{\ell}}$, or better its piecewise constant approximation with time-step h (an additional input, together with the signal duration T). As a first step, it computes recursively the quantitative satisfaction signals of subformula φ_1 for all locations $\ell \in L^{\hat{\ell}}_{[0,w_2]}$ and of subformula φ_2 for all locations $\ell \in L^{\hat{\ell}}_{[w_1,w_2]}$. Furthermore, it sets all the quantitative signals for φ_1 and φ_2 for the other locations to the constant signal equal to minus infinity. The algorithm runs a fixpoint computation for each time instant in the discrete time set $\{0, h, 2h, \ldots, mh\}$. The procedure is based on computing a function $\mathcal{X}$, with values in the extended reals $\mathbb{R}^*$, which is executed on the whole set of locations L, but for the modified signals equal to $-\infty$ for locations not satisfying the metric bounds for ℓ. The function $\mathcal{X}$ is defined below.

Definition 5. *Given a finite set of locations* L *and two functions* $s_1 : L \to \mathbb{R}^*, s_2 : L \to \mathbb{R}^*$*. The function* $\mathcal{X} : \mathbb{N} \times L \to \mathbb{R}$ *is inductively defined as:*

1. $\mathcal{X}(0, \ell) = s_1(\ell)$
2. $\mathcal{X}(i+1, \ell) = \min(\mathcal{X}(i, \ell), \min_{\ell' | \ell E \ell'}(\max(\mathcal{X}(i, \ell'), s_2(\ell'))))$

The algorithm then computes the function $\mathcal{X}$ iteratively, until a fixed-point is reached.

Theorem 1. *Let be* s_1 *and* s_2 *as in Definition 5, and*

$$s(\ell) = \max_{A \subseteq L, \ell \in A} (\min(\min_{\ell' \in A} s_1(\ell'), \min_{\ell' \in B^+(A)} s_2(\ell'))),$$

then

$$\lim_{i \to \infty} \mathcal{X}(i, \ell) = s(\ell), \quad \text{for all } \ell \in L.$$

Moreover, there exists $K > 0$ *such that* $\mathcal{X}(j, \ell) = s(\ell)$ *for all* $j \geq K$*.*

The following corollary provides the correctness of the method. It shows that, when $\mathcal{X}$ is computed for the modified signals constructed by the algorithm, it returns effectively the quantitative satisfaction score of the spatial surround.

Corollary 1. *Given an* $\hat{\ell} \in L$*, let* $\psi = \varphi_1 \mathcal{S}_{[w_1,w_2]} \varphi_2$ *and*

$$s_1(\ell) = \begin{cases} \rho(\varphi_1, \mathbf{x}, t, \ell) & \text{if } 0 \leq w(\hat{\ell}, \ell) \leq w_2 \\ -\infty & \text{otherwise.} \end{cases} \quad s_2(\ell) = \begin{cases} \rho(\varphi_2, \mathbf{x}, t, \ell) & \text{if } w_1 \leq w(\hat{\ell}, \ell) \leq w_2 \\ -\infty & \text{otherwise.} \end{cases}$$

Then $\rho(\psi, \mathbf{x}, t, \hat{\ell}) = s(\hat{\ell}) = \max_{A \subseteq L, \hat{\ell} \in A} (\min(\min_{\ell \in A} s_1(\ell), \min_{\ell \in B^+(A)} s_2(\ell)))$*.*

Algorithm 2 Quantitative monitoring for the surround operator

inputs: $\hat{\ell}, \psi = \varphi_1 \mathcal{S}_{[w_1,w_2]} \varphi_2$, h, T
for all $\ell \in L$ **do**
  if $0 \le w(\hat{\ell}, \ell) \le w_2$ then
    compute $s_{\varphi_1,\ell}$
    if $w(\hat{\ell}, \ell) \ge w_1$ then
      compute $s_{\varphi_2,\ell}$,
    else $s_{\varphi_2,\ell} = -\infty$
  else $s_{\varphi_1,\ell} = -\infty, s_{\varphi_2,\ell} = -\infty$
end for
for all $t \in \{0, h, 2h, \dots, T\}$ **do**
  for all $\ell \in L$ **do**
    $\mathcal{X}_{prec}(\ell) = +\infty$
    $\mathcal{X}(\ell) = s_{\varphi_1,\ell}(t)$
  end for
  while $\exists \ell \in L,$ s.t. $\mathcal{X}_{prec}(\ell) \ne \mathcal{X}(\ell)$ **do**
    $\mathcal{X}_{prec} = \mathcal{X}$
    for all $\ell \in L$ **do**
      $\mathcal{X}(\ell) = \min(\mathcal{X}_{prec}(\ell), \min_{\ell' | \ell E \ell'}(\max(s_{\varphi_2,\ell'}(t), \mathcal{X}_{prec}(\ell'))))$
    end for
  end while
  $s_{\psi,\hat{\ell}}(t) = \mathcal{X}(\hat{\ell})$
end for
return $s_{\psi,\hat{\ell}}$

In order to discuss the complexity of the monitoring procedure, we need an upper bound on the number of iterations of the algorithm. This is given by the following

Proposition 2. *Let d_G be the diameter of the graph G and $\mathcal{X}(\ell)$ the fixed point of $\mathcal{X}(i, \ell)$, then $\mathcal{X}(\ell) = \mathcal{X}(d_G + 1, \ell)$ for all $\ell \in L$.*

It follows that the computational cost for each location is $O(d_G |L| m)$, where m is the number of sampled time-points. The cost for all locations is therefore $O(d_G |L|^2 m)$.

4.1 Implementation

To support qualitative and quantitative monitoring of SSTL properties, a Java library has been developed. This library, named jSSTL[8], consists of three main packages: core, util and io. Package core provides the classes used to represent SSTL formulas. These classes mimic the *abstract syntax tree* of formulas. This package also includes the implementations of the monitoring algorithms presented in this section and of those previously introduced in [3].

Monitoring algorithms are implemented following the *visitor pattern*. Hence, monitoring is performed via a visit of a formula that implements a bottom-up evaluation. It is important to remark that the use of this pattern simplifies the integration of possible alternative monitoring algorithms. Each monitoring algorithm is rendered in terms of

[8] jSSTL is available on-line at `https://bitbucket.org/LauraNenzi/jsstl`

a class that is parametrised with respect to a *weighted graph* and provides the method check. The former represents the topology of the considered locations, while the latter takes as parameters an SSTL formula and a list of *piecewise constant signals* (one for each location) and returns a list of piecewise constant signals providing monitoring evaluation. The classes used to represent and manage *piecewise constant signals* are provided within package util. The implementation of weighted graphs relies on JGraphT[9]. This is a free Java graph library that provides mathematical graph-theory objects and algorithms.

Package io provides a set of classes that can be used to read graph models and input signals from an input stream and to write monitoring results to an output stream. Specific interfaces are also provided to simplify the integration of new specific input/output data formats. Currently, CSV and tabular based ascii files are supported for both input and output of signals.

5 Example: pattern formation in a reaction-diffusion system

In this section we show how SSTL can be used to identify the formation of *patterns* in a reaction-diffusion system. From the point of view of formal verification, the formation of patterns is an inherently spatio-temporal phenomenon, in that the relevant aspect is how the spatial organisation of the system changes over time. Alan Turing theorised in [16] that pattern formation is a consequence of the coupling of reaction and diffusion phenomena involving different chemical species, and can be described by a set of PDE reaction-diffusion equations, one for each species.

Our model, inspired by [11, 13], describes the production of skin pigments that generate spots in animal furs. The reaction-diffusion system is discretised, according to a Finite Difference scheme, as a system of ODEs whose variables are organised in a $K \times K$ rectangular grid. More precisely, we treat the grid as a weighted undirected graph, where each cell $(i, j) \in L = \{1, \dots, K\} \times \{1, \dots, K\}$ is a location (node), edges connect each pairs of neighbouring nodes along four directions (so that each node as at most 4 adjacent nodes), and the weight of each edge is always equal to the spatial length-scale δ of the system[10]. We consider two species A and B in a $K \times K$ grid, obtaining the system:

$$\begin{cases} \frac{dx^A_{i,j}}{dt} = R_1 x^A_{i,j} x^B_{i,j} - x^B_{i,j} + R_2 + D_1(\mu^A_{i,j} - x^A_{i,j}) & i = 1..,K,\ j = 1,..,K, \\ \frac{dx^B_{i,j}}{dt} = R_3 x^A_{i,j} x^B_{i,j} + R_4 + D_2(\mu^B_{i,j} - x^B_{i,j}) & i = 1..,K,\ j = 1,..,K, \end{cases} \tag{1}$$

where: $x^A_{i,j}$ and $x^B_{i,j}$ are the concentrations of the two species in the cell (i, j); R_i, $i = 1, ..., 4$ are the parameters that define the reaction between the two species; D_1 and D_2 are the diffusion constants; $\mu^A_{i,j}$ and $\mu^B_{i,j}$ are the inputs for the (i, j) cell, that is

$$\mu^n_{i,j} = \frac{1}{|\nu_{i,j}|} \sum_{\nu \in \nu_{i,j}} x^n_\nu \qquad n \in \{A, B\}, \tag{2}$$

[9] http://jgrapht.org

[10] For simplicity, here we fix $\delta = 1$. However, note that using a non-uniform mesh, for instance obtained by a Finite Elements approach in PDE integration, the weights of the edges of the resulting graph will not be uniform.

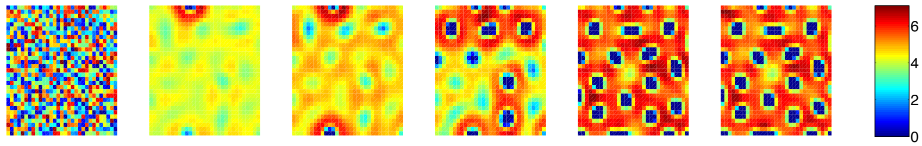

Fig. 1. Value of x^A for the system (1) for $t = 0, 5, 7, 12, 20, 50$ time units with parameters $K = 32, R_1 = 1, R_2 = -12, R_3 = -1, R_4 = 16, D_1 = 5.6$ and $D_2 = 25.5$. The initial condition has been set randomly. The colour map for the concentration is specified in the legend on the right.

where $v_{i,j}$ is the set of indices of cells adjacent to (i, j). The spatio-temporal trace of the system is the function $\mathbf{x} = (x^A, x^B) : [0, T] \times L \to \mathbb{R}^{K \times K} \times \mathbb{R}^{K \times K}$ where each x^A and x^B are the projection on the first and second variable, respectively. In Fig. 1, we report the concentration of A for a number of time points, generated by the numerical integration of System 1; at time $t = 20$ and $t = 50$, the shape of the pattern is apparent. Some regions can be identified, having a very low concentration of A (the spots) surrounded by regions with a very high concentration of A. The opposite happens for the concentration of B (high density regions surrounded by low density regions).

We will see now how we can use the surround operator to characterise the behaviour of this system. In order to classify spots, one should identify the sub-regions of the grid that present an high (or low) concentration of a certain species, surrounded by a low (high, respectively) concentration of the same species. Formally, one can e.g., capture the spots of the A species using the formula

$$\varphi_{spot} := (x^A \leq h) \mathcal{S}_{[w_1, w_2]} (x^A > h). \tag{3}$$

A trace $\mathbf{x}$ satisfies φ_{spot} at time t, in the location (i, j), $(\mathbf{x}, t, (i, j)) \models \varphi_{spot}$, if and only if there is a subset $L' \subset L$, that contains (i, j), such that all elements have a distance less than w_2 from (i, j), and x^A, at time t, is less or equal to h. Furthermore, each element in the boundary of this region has a concentration of A, at time t, greater than h, and its distance from (i, j) is between the interval $[w_1, w_2]$. Note that the use of distance bounds in the surround operator allows us to constrain the size/ diameter of the spot to $[w_1, w_2]$. Finally, combining the spatial property with temporal operators we can identify the insurgence time of the pattern and if it remains stable in the time:

$$\varphi_{pattern} := \mathcal{F}_{[T_{pattern}, T_{pattern}+\delta]} \mathcal{G}_{[0, T_{end}]} (\varphi_{spot}); \tag{4}$$

φ means that eventually in a time between $T_{pattern}$ and $T_{pattern} + \delta$ the property surround remains true for at least T_{end} time units. In Fig. 2 we show the validity of the property φ in each cell $(i, j) \in L$, for both the boolean and the quantitative semantics. Recalling that $(\mathbf{x}, \ell) \models \varphi$, if and only if $(\mathbf{x}, 0, \ell) \models \varphi$, the plots show the satisfaction at time $t = 0$. It is evident how well the procedure is able to identify which locations belong to the spots or not. If we make the distance constraint stricter, by reducing the width of the interval $[w_1, w_2]$, we are able to identify only the "centre" of the spot, as it is visible in Fig. 2 (d). However, in this case we may fail to identify spots that have an irregular shape (i.e., that deviate too much from a circular shape).

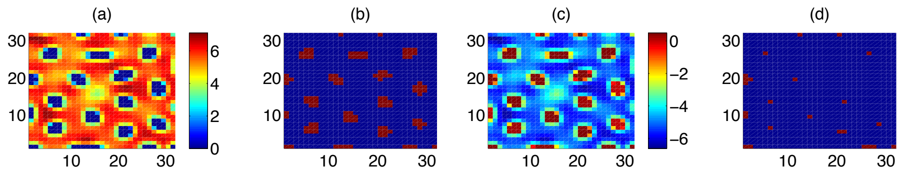

Fig. 2. Validity of the formula (4) with parameters $h = 0.5, T_{pattern} = 19, \delta = 1, T_{end} = 30, w_1 = 1, w_2 = 6$ for (b), (c) and $w_2 = 4$ for (d) . (a) Concentration of A at time t = 50; (b) (d) Boolean semantics of the property φ; the cells (locations) that satisfy the formula are in red, the others are in blue; (c) Quantitative semantics of the property φ; The value of the robustness is given by a colourmap as specified in the legend on the right of the figure.

Formula $\varphi_{pattern}$ describes the persistence of a spot in specific location. To describe a global spatial pattern, i.e. that every location has a nearby spot, we can express this in SSTL by the following formula:

$$\varphi_{ST-pattern} := \boxdot_{[0,w]} \diamondsuit_{[0,w']} \varphi_{pattern}, \tag{5}$$

where $\diamondsuit$ and $\boxdot$ are the everywhere and somewhere operators, w is chosen to cover all space, and w' measures the distance between spots. Checking this formula in a random location of our space is enough to verify the presence of the pattern.

Changing the diffusion constants D_1 and D_2 affects the shape and size of the spots or disrupts them (Fig. 3 (a)). In this case, we expect formula (5) to be false, and this is indeed the case. Formula (4), instead, is still true in some locations, due to particular boundary effects on the border of the grid (where fractions of spots still remain, as in Fig. 3 (a) right), or due the irregularity of the patterns (where, as Fig. 3 (a) left, some spots can have a shape similar to the model in Fig. 2 (a)).

A strength of spatio-temporal logics is the possibility to nest the temporal and spatial operators. We illustrate this in the following scenario. We assume as initial conditions of the system (1) its stable state, i.e. the concentrations of A and B at time 50 (see Fig. 2 (a)). However, we introduce a small perturbation, by changing a single value in a specific location in the centre of a spot. The idea is to study the effect of this perturbation, i.e. checking if it will disrupt the system or not. Specifically, we perturb the cell $(6,6)$, setting $x^A_{6,6}(0) = 10$. Dynamically, the perturbation is quickly absorbed and the system returns to the previous steady state. We formally investigate this scenario by checking the following property:

$$\varphi_p := (x^A \geq h_p) \wedge (\varphi_1 \mathcal{S}_{[w_m,w_M]} \varphi_2); \tag{6}$$

A trace $\mathbf{x}$ satisfies φ_p, in the location (i,j), if and only if $x^A_{i,j}(0) > h_p$ (the location is perturbed) and if there is a subset $L' \subseteq L$ that contains (i,j) such that all its elements have a distance less than w_M from (i,j) and satisfy $\varphi_1 = \mathcal{F}_{[0,T_p]}\mathcal{G}_{[0,T_d]}(x^A < h')$; φ_1 states that the perturbation of x^A is absorbed within T_p units of time, stabilising back to a value $x^A < h'$ for additional T_d time units. Furthermore, within distance $[w_m, w_M]$ from the original perturbation, where w_M is chosen such that we are within the spot of the non-perturbed system, $\varphi_2 := \mathcal{G}_{[0,T]}(x^A < h')$ is satisfied; i.e. no effect is observed, the value of x^A stably remains below h'. The meaning of φ_p is that the

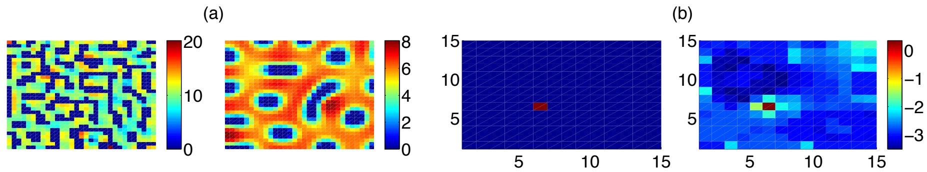

Fig. 3. (a) Snapshots at time $t = 50$ of x^A for the model (1) with parameters $D = [1.5, 23.6]$ (on the left) and $D = [8.5, 40.7]$ (on the right). (b) Boolean and quantitative semantics for the formula φ_p with parameters $w_m = 1$, $w_M = 2$, $T_p = 1$, $T_d = 10$ and $h' = 3$.

induced perturbation remains confined inside the original spot. In Fig. 3 (b) we report the evaluation of the quantitative semantics for φ_p, zooming in the 15×15 lower left corner of the original grid. All the locations that are not plotted have been evaluated and do not satisfy the property. As shown in the figure, the only location that satisfies this property is the perturbed one, $(6, 6)$.

The model (1) has been coded in Matlab/ Octave, and the monitoring has been performed by our Java implementation. Monitoring property $\varphi_{pattern}$ took 29.01 seconds for the boolean semantics and 67.85 seconds for the quantitative one (all locations and 100 time points), while property φ_p took 28,19 and 55,31 seconds, respectively. All the experiments were run on a Macbook Pro, OS X 10.9.5, Intel Core i5 processor with 2.6 GHz, 8GB 1600 MHz memory.

6 Discussion

We introduced the Signal Spatio-Temporal Logic, a spatio-temporal extension of STL [8], in which space is a finite metric structure represented by an undirected weighted graph. SSTL has the same temporal operators as STL, plus two spatial operators: the somewhere operator and the spatial surround operator. In SSTL, spatial and temporal operators can be arbitrarily nested. We provided the logic with a boolean and a quantitative semantics in the style of STL [8], and defined monitoring algorithms to evaluate such semantics on spatio-temporal trajectories. The monitoring procedures, implemented in Java, have been tested on a Turing reaction-diffusion system modelling a process of morphogenesis [16] in which spots are formed over time.

This work can be extended in several directions. First of all, we plan to perform a more thorough investigation of the expressivity of the logic, and to apply it on further case studies. In particular, we remark that SSTL can also be applied to describe properties of stochastic spatio-temporal systems, and the monitoring algorithms can be plugged in seamlessly into statistical model checking routines. Secondly, we plan to extend the definition of the logic itself to more general quasi-discrete metric spatial structures, exploiting the topological notion of closure spaces [4] and extending it to the metric case. Note that the current monitoring algorithms work already for more general spatial structures, like finite directed weighted graphs, but we plan to provide a more precise characterisation of the class of discrete spatial structures on which they can be applied. We will also optimise the implementation to improve performance, and additionally investigate if and how directionality can be expressed in SSTL. Finally, we plan to exploit the quantitative semantics for the robust design of spatio-temporal systems, along the lines of [2].

References

1. Marco Aiello, Ian Pratt-Hartmann, and Johan van Benthem, editors. *Handbook of Spatial Logics*. Springer, 2007.
2. E. Bartocci, L. Bortolussi, L. Nenzi, and G. Sanguinetti. System design of stochastic models using robustness of temporal properties. *Theoretical Computer Science*, 2015.
3. L. Bortolussi and L. Nenzi. Specifying and monitoring properties of stochastic spatio-temporal systems in signal temporal logic. In *Proceedings of VALUETOOLS*, 2014.
4. V. Ciancia, D. Latella, M. Loreti, and M. Massink. Specifying and verifying properties of space. In *Proceedings of IFIP-TCS*, number 8705 in LNCS, pages 222–235, 2014.
5. Vincenzo Ciancia, Stephen Gilmore, Gianluca Grilletti, Diego Latella, Michele Loreti, and Mieke Massink. Spatio-temporal model-checking of vehicular movement in public transport systems. *Submitted*, 2015.
6. Vincenzo Ciancia, Stephen Gilmore, Diego Latella, Michele Loreti, and Mieke Massink. Data verification for collective adaptive systems: Spatial model-checking of vehicle location data. In *Proceedings of SASOW, 2014*.
7. A. Donzé, T. Ferrer, and O. Maler. Efficient robust monitoring for stl. In *Proc. of CAV'13*, volume 8044 of *LNCS*, pages 264–279, 2013.
8. A. Donzé and O. Maler. Robust satisfaction of temporal logic over real-valued signals. In *Proc. of FORMATS*, volume 6246, pages 92–106, 2010.
9. Georgios E. Fainekos and George J. Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410(42):4262–4291, September 2009.
10. Antony Galton. The mereotopology of discrete space. In Christian Freksa and DavidM. Mark, editors, *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science*, volume 1661 of *Lecture Notes in Computer Science*, pages 251–266. Springer Berlin Heidelberg, 1999.
11. E. Aydin Gol, E. Bartocci, and C. Belta. A formal methods approach to pattern synthesis in reaction diffusion systems. In *Proceedings of CDC*, 2014.
12. R. Grosu, E. Bartocci, F. Corradini, E. Entcheva, S. A. Smolka, and A. Wasilewska. Learning and detecting emergent behavior in networks of cardiac myocytes. In *Proceedings of HSCC'08*, pages 229–243, 2008.
13. I. Haghighi, A. Jones, J. Z. Kong, E. Bartocci, Grosu R., and C. Belta. SpaTeL: A Novel Spatial-Temporal Logic and Its Applications to Networked Systems. In *Proceedings of HSCC*, 2015.
14. O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *Proceedings FORMATS 2004*, volume 3253, pages 152–166, 2004.
15. L. Mari, E. Bertuzzo, L. Righetto, R. Casagrandi, M. Gatto, I. Rodriguez-Iturbe, and A. Rinaldo. Modelling cholera epidemics: the role of waterways, human mobility and sanitation. *Journal of The Royal Society Interface*, 9(67):376–388, February 2012.
16. A. M. Turing. The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 237(641):37–72, August 1952.

A Proofs

In this appendix, we present the proofs of Proposition 1 and 2, Theorem 1 and Corollary 1.

Proposition 1. *Let the primary signal* $\mathbf{x}$ *be Lipschitz continuous, as the functions defining the atomic predicates. Let* K *be a Lipschitz constant for all secondary signals, and* h *be the discretisation step. Given a SSTL formula* φ*, let* $u(\varphi)$ *counts the number of temporal until operators in* φ*, and denote by* $\rho(\varphi, \mathbf{x})$ *its satisfaction score over the trace* $\mathbf{x}$ *and by* $\rho(\varphi, \hat{\mathbf{x}})$ *the satisfaction score over the discretised version* $\hat{\mathbf{x}}$ *of* $\mathbf{x}$ *with time step* h*. Then*

$$\|\rho(\varphi, \mathbf{x}) - \rho(\varphi, \hat{\mathbf{x}})\| \leq u(\varphi) K h$$

Proof. We first observe that the monitoring algorithm for boolean and spatial operators preserve the error of the input quantitative signals. This means that if $\|s_{\varphi_j,\ell} - \hat{s}_{\varphi_j,\ell}\| \leq \varepsilon$, then $\|s_{\psi,\ell} - \hat{s}_{\psi,\ell}\| \leq \varepsilon$, for ψ one of $\neg\varphi_1$, $\varphi_1 \wedge \varphi_2$, $\varphi_1 \mathcal{S}_{[w_1,w_2]} \varphi_2$, $\diamondsuit_{[w_1,w_2]} \varphi_1$. Hence, temporal discretisation introduces errors only for temporal operators.

Now, let $I = [t_1, t_2]$ be such that $t_j = k_j h$, and denote the Minkowski sum by $\oplus$, so that $t \oplus I = [t + t_1, t + t_2]$. Denote by $\hat{I}$ the discretised version of I, with step h, $\hat{I} = \{k_1 h, (k_1 + 1)h, \ldots, k_2 h\}$. We observe two facts for the maximum, with identical statements holding for the minimum.

- Let $f(t)$ be Lipschitz with constant K. Let $g(t) = \max_{t' \in t \oplus I} f(t)$ and $\hat{g}(t) = \max_{t' \in t \oplus \hat{I}} f(t)$. Then $\|g(t) - \hat{g}(t)\| \leq Kh/2$. This holds by applying the Lipschitz property between a generic point in $t \oplus I$ and the closest point in $t \oplus \hat{I}$, and noting that the maximum distance between such points is $h/2$.
- If $\tilde{f}$ is such that $\|\tilde{f}(t) - f(t)\| \leq \varepsilon$ uniformly in t, and we let g, $\hat{g}$ as above, and $\tilde{g}(t) = \max_{t' \in t \oplus \hat{I}} \tilde{f}(t)$, then

$$\|g(t) - \tilde{g}(t)\| \leq \|g(t) - \hat{g}(t)\| + \|\hat{g}(t) - \tilde{g}(t)\| \leq Kh/2 + \varepsilon.$$

Hence, the second property implies that the additional error we introduce by evaluating a time bounded until is an additive term no larger than Kh, as in the definition of the quantitative semantics of the until, there are a nested minimum and a maximum over dense time intervals. Hence the total error will be bounded by Kh times the number of temporal operators. ∎

Theorem 1. *Let* s_1 *and* s_2 *be as in Definition 5, and*

$$s(\ell) = \max_{A \subseteq L, \ell \in A} (\min(\min_{\ell' \in A} s_1(\ell'), \min_{\ell' \in B^+(A)} s_2(\ell')))$$

then

$$\lim_{i \to \infty} \mathcal{X}(i, \ell) = s(\ell), \quad \textit{for all } \ell \in L.$$

Moreover, there exists $K > 0$ *such that* $\mathcal{X}(j, \ell) = s(\ell)$ *for all* $j \geq K$*.*

Note that s is equivalent to the quantitative semantics of the surround operator $\varphi_1 \mathcal{S} \varphi_2$, with s_i denoting the robustness of φ_i, without the distance constraints. We first present two lemmas, followed by the proof of Theorem 1.

Lemma 1. *If $\mathcal{X}(k+1,\ell) = \mathcal{X}(k,\ell)$ for all $\ell \in L$ then, $\forall i > k$, $\mathcal{X}(i,\ell) = \mathcal{X}(k,\ell)$.*

Proof. By induction.

- *(basis step) i=k +1 is true by hipothesis,*
- *(inductive step) suppose the assert holds for $i > k$, i.e. $\mathcal{X}(i,\ell) = \mathcal{X}(k,\ell)$ (I.H.), then we have to prove that it holds for $i+1$.*

$$\begin{aligned}
\mathcal{X}(i+1,\ell) &= \min(\mathcal{X}(i,\ell), \min_{\ell' | \ell E \ell'}(\max(\mathcal{X}(i,\ell'), s_2(\ell')))) && \{\textit{by Def. of } \mathcal{X}\} \\
&= \min(\mathcal{X}(k,\ell), \min_{\ell' | \ell E \ell'}(\max(\mathcal{X}(k,\ell'), s_2(\ell')))) && \{\textit{by I.H.}\} \\
&= \mathcal{X}(k+1,\ell) = \mathcal{X}(k,\ell). && \{\textit{by Def. of } \mathcal{X}\}
\end{aligned}$$

■

Lemma 2. *Let A_ℓ be the subregion that maximizes $s(\ell)$, then, $\forall \ell' \in A_\ell$, $s(\ell') \geq s(\ell)$.*

Proof. If A_ℓ is the subregion that maximizes $s(\ell)$ then

$$s(\ell) = \min(\min_{\ell' \in A_\ell} s_1(\ell'), \min_{\ell' \in B^+(A_\ell)} s_2(\ell')))$$

Suppose by contradiction that $\exists \hat{\ell} \in A_\ell$ s.t. $s(\hat{\ell}) < s(\ell)$. Let $Q = \{A \subseteq L, \hat{\ell} \in A\}$. This means that

$$s(\hat{\ell}) < s(\ell)$$
$$\equiv$$
$$\max_{A \in Q}(\min(\min_{\ell' \in A} s_1(\ell'), \min_{\ell' \in B^+(A)} s_2(\ell'))) < \min(\min_{\ell' \in A_\ell} s_1(\ell'), \min_{\ell' \in B^+(A_\ell)} s_2(\ell')))$$

But A_ℓ is a subset of L and $\hat{\ell} \in A_\ell$ therefore $A_\ell \in Q$, thus the inequality can not hold. ■

Proof (of Theorem 1). We have to prove that (1) $\mathcal{X}(i,\ell)$ converges in a finite number of steps, in each location ℓ, to $\mathcal{X}(\ell) \in \mathbb{R}^*$ and that (2) $\forall \ell \in L$, $\mathcal{X}(\ell) = s(\ell)$.

1. Convergence of $\mathcal{X}$.
   First note that $\mathcal{X}(i,\ell) \geq \min(\mathcal{X}(i,\ell), \min_{\ell' | \ell E \ell'}(\max(\mathcal{X}(i,\ell'), s_2(\ell')))) = \mathcal{X}(i+1,\ell)$, thus $\mathcal{X}_{|\ell}$ is a monotonic decreasing function. Second, note that $\mathcal{X}(i,\ell) \in \{s_j(\ell) \mid j \in \{1,2\}, \ell \in L\}$ is a finite set of sortable values. So, in every step, $\mathcal{X}$ takes a value of a sortable finite set. Finally, if it happens that for a step, for all $\ell \in L$, $\mathcal{X}(i,\ell)$ does not change then, from Lemma 1, it will remain the same for all the next steps. The convergence of $\mathcal{X}$ to the maximum fixed point follows then from Tarsky's theorem.

2. We have to prove that $\forall \ell$, $\mathcal{X}(\ell) = s(\ell)$.
   Let A_ℓ be the subregion that maximizes $s(\ell)$ then

$$s(\ell) = \min(\min_{\ell' \in A_\ell} s_1(\ell'), \min_{\ell' \in B^+(A_\ell)} s_2(\ell'))).$$

   First we prove that $\forall \ell$, $\mathcal{X}(\ell) \geq s(\ell)$ (2a) and then that they are equal (2b).

   2a) To prove that $\mathcal{X}(\ell) \geq s(\ell)$ it suffices to prove that, for a generic ℓ, $\forall i \in \mathbb{N}$, $\mathcal{X}(i,\ell) \geq s(\ell)$, and for the convergence of $\mathcal{X}$ that $\exists j \in \mathbb{N}$ s.t. $\mathcal{X}(\ell) = \mathcal{X}(j,\ell), \forall \ell, \forall j \geq i$. The proof is by induction.
   - (basis step)

$$\mathcal{X}(0,\ell) = s_1(\ell) \geq \min_{\ell' \in A_\ell} s_1(\ell') \geq \min(\min_{\ell' \in A_\ell} s_1(\ell'), \min_{\ell' \in B^+(A_\ell)} s_2(\ell'))) = s(\ell)$$

   - (inductive step) Assume $\mathcal{X}(i,\ell) \geq s(\ell)$, to prove that $\mathcal{X}(i+1,\ell) \geq s(\ell)$;

$$\mathcal{X}(i+1,\ell) = \min(\mathcal{X}(i,\ell), \min_{\ell' | \ell E \ell'} (\max(\mathcal{X}(i,\ell'), s_2(\ell')))) \tag{7}$$

$$\geq \min_{\ell' R \ell}(\min_{\ell' \in A_\ell}(\max \mathcal{X}(i,\ell'), s_2(\ell')), \min_{\ell' \in B^+(A_\ell)}(\max \mathcal{X}(i,\ell'), s_2(\ell'))) \tag{8}$$

$$\geq \min_{\ell' R \ell}(\min_{\ell' \in A_\ell} \mathcal{X}(i,\ell'), \min_{\ell' \in B^+(A_\ell)} s_2(\ell')) \tag{9}$$

     - if $\ell' \in A_\ell$, then $\mathcal{X}(i,\ell') \geq s(\ell') \geq s(\ell)$. The first inequality is true due to the I.H., the second because of Lemma 2;
     - if $\ell' \in B^+(A_\ell)$, then $s_2(\ell') \geq s(\ell)$ because of the definition of s;

   2b) Suppose by contradiction that $\exists \hat{\ell} \in L$ s.t. $\mathcal{X}(\hat{\ell}) > s(\hat{\ell})$. At the fixed point we have that

$$\mathcal{X}(\hat{\ell}) = \min(\mathcal{X}(\hat{\ell}), \min_{\ell | \hat{\ell} E \ell}(\max(\mathcal{X}(\ell), s_2(\ell))))$$

   This means that the inequality

$$\min_{\ell | \hat{\ell} E \ell}(\max(\mathcal{X}(\ell), s_2(\ell))) > s(\hat{\ell}) \tag{10}$$

   has to be true.
   Let $A \subseteq L$, we define:
   - $C(A) \coloneqq \{\ell \in L \big| \exists \ell' \in A \text{ s.t. } \ell' E \ell \wedge \mathcal{X}(\ell) \geq s_2(\ell)\}$
   - $C^i(A) = C(C^{i-1}(A))$

   We can then define the closure of C, as $C^*(A) = A \bigcup_{i=0}^{\infty} C^i(A)$.
   Because of the definition of C and the inequality (10) we have that $\forall \ell \in C^*(\{\hat{\ell}\})$, $s_1(\ell) \geq \mathcal{X}(\ell) > s(\hat{\ell})$ and that $\forall \ell \in B^+(C^*(\{\hat{\ell}\}))$, $s_2(\ell) > s(\hat{\ell})$, so

$$\min(\min_{\ell \in C^*(\{\hat{\ell}\})} s_1(\ell), \min_{\ell \in B^+(C^*(\{\hat{\ell}\}))} s_2(\ell))) > s(\hat{\ell})$$

i.e.

$$\min(\min_{\ell \in C^*(\{\hat{\ell}\})} s_1(\ell), \min_{\ell \in B^+(C^*(\{\hat{\ell}\}))} s_2(\ell))) > \min(\min_{\ell \in A_{\hat{\ell}}} s_1(\ell), \min_{\ell' \in B^+(A_{\hat{\ell}})} s_2(\ell')))$$

but this contradicts the assumption of maximality of $A_{\hat{\ell}}$. ∎

In the following the distance constraints are addressed.

Corollary 2. *Given an $\hat{\ell} \in L$, let $\psi = \varphi_1 \mathcal{S}_{[w_1,w_2]} \varphi_2$ and*

$$s_1(\ell) = \begin{cases} \rho(\varphi_1, \mathbf{x}, t, \ell) & \textit{if } 0 \le w(\hat{\ell}, \ell) \le w_2 \\ -\infty & \textit{otherwise}. \end{cases}$$

$$s_2(\ell) = \begin{cases} \rho(\varphi_2, \mathbf{x}, t, \ell) & \textit{if } w_1 \le w(\hat{\ell}, \ell) \le w_2 \\ -\infty & \textit{otherwise}. \end{cases}$$

Then $\rho(\psi, \mathbf{x}, t, \hat{\ell}) = s(\hat{\ell}) = \max_{A \subseteq L, \hat{\ell} \in A} (\min(\min_{\ell \in A} s_1(\ell), \min_{\ell \in B^+(A)} s_2(\ell)))$.

Proof. We recall that

$$\rho(\psi, \mathbf{x}, t, \hat{\ell}) = \max_{A \subseteq L^{\hat{\ell}}_{[0,w_2]}, \ell \in A, B^+(A) \subseteq L^{\hat{\ell}}_{[w_1,w_2]}} (\min(\min_{\ell \in A} \rho(\varphi_1, \mathbf{x}, t, \ell), \min_{\ell \in B^+(A)} \rho(\varphi_2, \mathbf{x}, t, \ell))).$$

where $L^{\hat{\ell}}_{[w_1,w_2]} := \{\ell \in A | w_1 \leqslant w(\ell, \hat{\ell}) \le w_2\}$. This means that $\ell \in A$ iff $w(\ell, \hat{\ell}) \le w_2$ and, for all $\ell' E \ell$, $w_1 \leqslant w(\ell', \hat{\ell}) \le w_2$.

So, we consider a restricted number of subsets of L for ρ and all the possible subsets of L for s. Furthermore, the value of the locations considered by both are always the same, i.e. the value of s_1 and s_2 differ only in the locations considered by s and not by ρ. For this reason $s(\ell) \ge \rho(\ell)$.

Let A_ρ be the subset that maximizes ρ of $\hat{\ell}$ and A_s the subset that maximizes s of $\hat{\ell}$. And suppose by contradiction that

$$\min(\min_{\ell \in A_s} s_1(\ell), \min_{\ell' \in B^+(A_s)} s_2(\ell))) > \min(\min_{\ell \in A_\rho} \rho(\varphi_1, \mathbf{x}, t, \ell), \min_{\ell \in B^+(A_\rho)} \rho(\varphi_2, \mathbf{x}, t, \ell))),$$

but the values considered by s and not by ρ are all equal to $-\infty$ (see line 8 of Alg. 2), so if A_s has a location that cannot be considered by ρ it means that

$$\min(\min_{\ell \in A_s} s_1(\ell), \min_{\ell' \in B^+(A_s)} s_2(\ell))) = -\infty$$

but minus infinity cannot be bigger than any number. ∎

Proposition 2. *Let d_G be the diameter of the graph G and $\mathcal{X}(\ell)$ the fixed point of $\mathcal{X}(i, \ell)$, then $\mathcal{X}(\ell) = \mathcal{X}(d_G + 1, \ell)$ for all $\ell \in L$.*

Proof. The graph diameter of G is equal to $d_g = \max_{\ell,\ell' \in L} d(\ell,\ell')$. Recall that $\mathcal{X}(d_g,\ell) \in \{s_j(\ell) \mid j \in \{1,2\}, \ell \in L\}$ is a finite set of sortable values. At step zero the value of $\mathcal{X}$ is equal to s_1 in all the locations. At each next step, the value of $\mathcal{X}(i,\ell)$ depends only on the value of $\mathcal{X}$ in the same location at the previous step and the value of s_2 and $\mathcal{X}$ in the previous step in the direct neighbours of ℓ, $\ell' E \ell$. This means that, after a number of steps equal to the diameter of the graph, i.e. the longest shortest path of the network, $\mathcal{X}$, for all nodes ℓ, has taken into account the values s_1 and s_2 of all the nodes. ∎