A spatial logic with time and quantifiers*

Laura Bussi, Vincenzo Ciancia, and Fabio Gadducci

Department of Computer Science, University of Pisa, Italy
laura.bussi@phd.unipi.it, fabio.gadducci@unipi.it
CNR-ISTI, Pisa, Italy
l.bussi@isti.cnr.it, v.ciancia@isti.cnr.it

Abstract. Spatial logics are formalisms for expressing topological properties of structures based on geometrical entities and relations. In this paper we consider SLCS, the Spatial Logic for Closure Spaces, recently used for describing features of images and video frames. We extend SLCS in two directions. We first introduce first-order quantifiers, ranging on both individuals and atomic propositions. We then equip the logic with temporal operators, and provide a linear-time semantics over finite traces. The resulting formalism allows to state properties about geometrical entities whose attributes change along time. For both extensions, we prove the equivalence of their operational semantics with a denotational one.

1 Introduction

Spatial logics are formalisms for expressing topological properties of structures based on geometrical entities and relations, and as such have been extensively studied since the first half of the last century [1]. Recently, such logics have been further explored for the modelling of computational devices, ranging from collective adaptive [13, 14] and cyber-physical systems [24, 22] to pattern synthesis [5].

Introduced in [16], the Spatial Logic for Closure Spaces (SLCS) uses as models a generalisation of topological spaces, known as *pretopological* or *Čech closure spaces*. These spaces include interesting structures such as binary relations/simple graphs. And since images can be interpreted as graphs, whose structure is given by pixels with a chosen adjacency relation, the SLCS model checker `VoxLogicA` [7] has been used for the analysis of 2D/3D pictures, in particular for the problem of "contouring" in medical imaging [4, 6].

SLCS has proved to be quite expressive in characterising the structural properties of a graph. However, it does not possess operators for constructing *named references* to "individuals" — be these points, regions, atomic propositions, or agents moving in space. For instance, one might ask if there is a region X of an

* Supported by University of Pisa project PRA_2022_99 "FM4HD", MUR project PRIN 20228KXFN2 "STENDHAL", CNR (Italy) and SRNSFG (Georgia) bilateral project CNR-22-010 "Model Checking for Polyhedral Logic", and European Union - Next Generation EU - MUR project PNRR PRI ECS00000017 PRR.AP008.003 "THE - Tuscany Health Ecosystem". The authors thank Diego Latella and Mieke Massink for fruitful discussions on spatio-temporal logics and their applications.

image, satisfying a given logical property, which in some time will become larger than another one. This kind of analysis has immediate applications in medical imaging for *lesion tracking*, focussing on the temporal evolution of a lesion in a series of snapshots of a patient's situation (a "longitudinal study"). In this work, we develop the ideas of [16] and [10], adopting the same setting of [17] to model spatio-temporal situations. First of all, we provide a precise correspondence between spaces and relations, streamlining various results discussed in the literature on SLCS. We also present a succinct syntax of SLCS, including just the backward $\overleftarrow{\rho}$ and forward $\vec{\rho}$ reachability operators, which reflect the well-known *until* operator of temporal logic and have efficient model checking algorithms in `VoxLogicA`. Such operators allow to state properties of points of space akin to *there is a finite path from point x_1 to some point x_2, such that x_2 satisfies a given formula ϕ_2, and the path passes only through points satisfying another formula ϕ_1*. Taking inspiration from [10], we introduce two extensions of SLCS. The first one concerns first-order quantification, which may predicate on points of a space and the atomic propositions they may satisfy. The second introduces temporal operators, similar in spirit to [14]. Finally, these extensions are merged, distilling an expressive and flexible quantified spatio-temporal logic.

A running example: video stream analysis. The logic we propose allows to state properties involving the identity of a node, in a graph whose structure does not change, yet the propositions holding at each node may. Throughout the paper, we illustrate its expressiveness by a simple example: the analysis of video streams, demonstrated using the well-known Pac-Man™ videogame. The example is taken from [11], where only purely spatial properties were considered.

Pac-Man is a 2D video game released by the Japanese firm Bandai-Namco in 1980. It has a simple, yet interesting structure: the main character of the game, Pac-Man, moves inside a maze. Along the corridors, several peach dots are placed, together with four energiser pellets positioned in the corners. Furthermore, four coloured ghosts (Inky, Blinky, Pinky, and Clyde) try to capture Pac-Man, moving in the maze according to different routines. A twist happens when Pac-Man eats an energiser pellet: in this case, the ghosts' colours turn to blue, and they can be caught by Pac-Man instead. The aim of a single level is to eat all the dots and pellets, avoiding to be captured by a ghost.

Despite its simplicity, the Pac-Man videogame is a clear example of applicability of our logical framework. The spatial structure does not change along time: the graph underlying each video frame is always the same. Instead, atomic properties associated to a node/pixel, that is, the colours, vary along time: for example, Pac-Man is represented by yellow-coloured pixels that are inside the maze (note that there are other areas with the same colour, representing the remaining lives, see Figure 1). Such a setting is useful in real-world applications. Consider, for instance, lesion tracking in medical imaging. The input data are *snapshots* of a patient at different times. After what is called the *co-registration* phase, all images have the same structure (resolution and physical dimensions). In other words, the underlying graph never changes, while the colours of the pixels, i.e. the atomic propositions, change along the temporal axis.

Related work The task to investigate quantification in modal logic interpreted over spaces was already tackled in various works. An important example are the works by Awodey and Kishida [2, 21], where first order modal logic is provided with a topological interpretation. The proposed approach is quite different from ours: in this case, sheaves are used to combine denotational semantics of modal logic and first order logic, and quantification is permitted only over points. Moreover, this approach applies only to topological spaces.

Spatio-temporal reasoning has also been a topic of interest along years, and various approaches have been proposed to combine space and time. Products of modal logics have been considered to this end [8]. Products of modal logics give rise to *multi-modal* logic languages, where different modal operators can be used to reason about different aspects of a model (in this case, the spatial and temporal aspects). Despite the fact that we also consider products of modal logics, the cited proposal is quite different. Again in this case, only topological structures are considered, and the temporal fragment is interpreted over the pair $(\mathbb{N}, <)$, thus being equivalent to the classic PTL temporal logic. In our case, instead, we only consider interpretation over finite traces. A comprehensive study of spatio-temporal approaches to modal logics is given by [17], where various kinds of spaces (e.g. Euclidean or Aleksandroff) are considered. This work offers an interesting study of the tradeoff between expressivity and complexity of various spatio-temporal logic, and it is our main reference for state-of-the-art languages that combine space and time. Still, the topic of the considered logics is topological spaces, thus lacking the generality that we aim to have.
Closer to our proposal, and in some sense orthogonal to it, is the one developed in [14], where branching time operator where introduced and no quantification was considered. In this case, the language was developed to reason about evolving smart systems (e.g. bike sharing systems), thus a branching time logic was adopted for the temporal part. We drop this kind of approach in favour of linear time operators, which are more likely to be useful in a setting of medical imaging, where we state properties about a set of images on a single timeline.

Synopsis. The structure of the paper follows. Section 2 gives an overview of the models currently used for SLCS and we recast them uniformly, making precise the correspondence with binary relations/simple graphs. Section 3, presents a succinct version of SLCS, which is equipped with existential quantifiers in Section 4 and with linear-time operators in Section 5. Finally, Section 6 proposes a quantified spatio-temporal logic. Each section gives the correspondence between the semantics with respect to a single spatial path/temporal trace and a denotational one, and it is rounded up with an instance of our running example. Section 7 closes the paper, summing up our results and hinting at future works.

2 Some notions on spaces and relations

We recall some notions related to spaces, used as domains of interpretation of various logics (see [1]) including SLCS, and discuss their links with binary relations/simple graphs, making precise remarks scattered in papers on SLCS.

2.1 Preliminaries on spaces

We open by listing some basic properties and definitions for spaces.

Definition 1. *A space $\mathcal{C}$ is a pair (S, C) such that S is a set of points and $C : 2^S \to 2^S$ is a function satisfying $C(\emptyset) = \emptyset$ and $C(X \cup Y) = C(X) \cup C(Y)$ for $X, Y \subseteq S$. A space is complete if $C(\bigcup_{i \in I} X_i) = \bigcup_{i \in I} C(X_i)$ for any I.*

If S is finite then a space (S, C) is always complete. Given a space (S, C) and a subset $X \subseteq S$, we denote the complement $S \setminus X$ of X in S as X^c. And while C is called the closure operator, its dual is the interior $I(X) = C(X^c)^c = S \setminus C(S \setminus X)$.

Definition 2. *A space (S, C) is* pre-topological *if $X \subseteq C(X)$ holds for all $X \subseteq S$; it is* Alexandrov *if it is pre-topological and complete; and it is* topological *if it is pre-topological and $C(C(X)) \subseteq C(X)$ holds for all $X \subseteq S$.*

The notions above are standard from the literature on topology. In the literature on spatial logics, pre-topological and Alexandrov spaces are called Cêch closure spaces and quasi-discrete Cêch closure spaces, respectively.

Note that for any space we can define a sort of inverse $\mathcal{C}^{-1} = (S, C^{-1})$, for $C^{-1}(X) = \bigcup_{x \in X} \{y \mid x \in C(\{y\})\}$, which is complete by definition. In order to identify those cases where a space and its inverse interact properly, we take inspiration from modal algebras and introduce the notion of conjugate spaces.

Definition 3. *Two spaces (S, C_1) and (S, C_2) are* conjugate *if they satisfy $X \subseteq I_1(C_2(X)) \cap I_2(C_1(X))$.*

Remark 1. The law for conjugate spaces can be stated as "$C_1(X) \subseteq Y$ iff $X \subseteq C_2(Y)$", which explicitly tells that the two closures are the respective inverses.

Proposition 1. *Let $\mathcal{C}$ be a complete space. Then $\mathcal{C}$ and $\mathcal{C}^{-1}$ are conjugate.*

Proof. We just need to prove that for any X, Y we have that $C(X) \cap Y = \emptyset$ iff $X \cap C^{-1}(Y) = \emptyset$. Now, let us assume that $C(X) \cap Y = \emptyset$ and there exists x such that $x \in X \cap C^{-1}(Y)$. Thus $x \in X$ and $x \in C^{-1}(Y)$. By definition, $x \in C^{-1}(Y)$ implies that there exists $y \in Y$ such that $x \in C^{-1}(\{y\})$, that is, $y \in C(\{x\})$, hence $y \in C(X)$ since C is complete, thus $y \in C(X) \cap Y$, a contradiction. The inverse direction is analogous.

Remark 2. Note that we cannot drop the completeness requirement for $\mathcal{C}$ in the proposition above. Consider e.g. the set $\mathbb{N}$ of natural numbers and a function $C : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ such that $C(X) = \emptyset$ if X is either empty or finite, and $C(X) = \mathbb{N}$ if X is infinite. Clearly, $(\mathbb{N}, C)$ is a space, albeit not complete. Now, we have that $C(\{n\}) = \emptyset$ for all $n \in \mathbb{N}$, so that $C^{-1}(\{m\}) = \{n \mid m \in C(\{n\})\} = \emptyset$ for all $m \in \mathbb{N}$, which implies that $C^{-1}(Y) = \emptyset$ for all $Y \subseteq \mathbb{N}$. Thus, for any infinite set $X \subseteq \mathbb{N}$, we have that $C(X) \cap Y = Y$ while $X \cap C^{-1}(Y) = \emptyset$.

2.2 Spaces vs. relations

There is a reason to focus on complete spaces, namely, the fact that they have a tight connection with binary relations (i.e. simple graphs/unlabelled Kripke frames). In the following we consider relations on a set S: we identify them as functions $R : S \to 2^S$ and denote $2^R : 2^S \to 2^S$ the lifting $2^R(X) = \bigcup_{x \in X} R(x)$.

Now, each space $\mathcal{C} = (S, C)$ induces a relation $R_{\mathcal{C}} : S \to 2^S$ defined as $R_{\mathcal{C}}(x) = C(\{x\})$. Note that for any finite $X \subseteq S$ it holds $2^{R_{\mathcal{C}}}(X) = C(X)$, and the equality holds also for infinite X if $\mathcal{C}$ is complete. Vice versa, each relation $R : S \to 2^S$ induces a complete space $\mathcal{C}_R = (S, C_R)$ defined as $C_R(X) = 2^R(X)$.

Lemma 1. *Let $R : S \to 2^S$ be a relation. Then $R_{\mathcal{C}_R}(x) = R(x)$ for all $x \in S$. Let $\mathcal{C}$ be a complete space. Then $\mathcal{C}_{R_{\mathcal{C}}}(X) = C(X)$ for all $X \subseteq S$.*

Thus, interpreting logics on complete spaces is the same as using as models the underlying relations. What is also noteworthy is that some laws holding for complete spaces turn out to state structural properties of such relations.

Proposition 2. *Let $\mathcal{C}$ be a complete space and $R_{\mathcal{C}}$ the associated relation. Then*

- *$\mathcal{C}$ satisfies $X \subseteq C(X)$ iff $R_{\mathcal{C}}$ is reflexive*
- *$\mathcal{C}$ satisfies $C(C(X)) \subseteq C(X)$ iff $R_{\mathcal{C}}$ is transitive*
- *$\mathcal{C}$ satisfies $X \subseteq I(C(X))$ iff $R_{\mathcal{C}}$ is symmetric*

Proof. The first two items are kind of obvious thanks to Proposition 1. Thus, let us now look at the third property. For $R_{\mathcal{C}}$ being symmetric means that for all x, y it holds that $y \in R_{\mathcal{C}}(x)$ iff $x \in R_{\mathcal{C}}(y)$ or, equivalently, that $y \notin R_{\mathcal{C}}(x)$ iff $x \notin R_{\mathcal{C}}(y)$. Satisfying $X \subseteq I(C(X))$ means that $X \subseteq C(C(X)^c)^c$. Recall now that for a complete space we have $2^{R_{\mathcal{C}}}(X) = C(X)$, and for the sake of calculations consider the relation $D(x) = S \setminus R_{\mathcal{C}}(x)$. Thus, axiom $X \subseteq I(C(X))$ can be expressed as $X \subseteq C(\bigcap_{x \in X} D(x))^c = \bigcap_{z \in \bigcap_{x \in X} D(x)} D(z)$.

($\Longrightarrow$) Let us assume that there exist x, y such that $x \in R_{\mathcal{C}}(y)$ and $y \in D(x)$. Assuming $X = \{x\}$, the axiom becomes $x \in \bigcap_{z \in D(x)} D(z)$. Since $y \in D(x)$, the axiom implies $x \in D(y)$, which contradicts $x \in R_{\mathcal{C}}(y)$.

($\Longleftarrow$) Let us assume that $R_{\mathcal{C}}$ is symmetric and that there exists X such that $X \nsubseteq I(C(X))$. The latter means that there exists $y \in X$ such that $y \notin I(C(X))$. So, there exists $w \in \bigcap_{x \in X} D(x)$ such that $y \notin D(w)$, i.e. $y \in R_{\mathcal{C}}(w)$. By symmetry $w \in R_{\mathcal{C}}(y)$, that is, $w \notin D(y)$, which contradicts $w \in \bigcap_{x \in X} D(x)$.

Finally, recall how for a space (S, C) we defined a kind of inverse space (S, C^{-1}), inspired by the analogous notion for relations: in fact, given $R : S \to 2^S$, its inverse $R^{-1} : S \to 2^S$ is the relation such that $R^{-1}(x) = \{y \mid x \in R(y)\}$.

Proposition 3. *Let (S, C) be a space. Then $R_C^{-1} = R_{C^{-1}}$.*

3 Spatial logics

This section recalls syntax and semantics of spatial logics (SL), introduces its denotational semantics, and makes precise its connection with CTL.

We start by assuming a set P of atomic propositions, ranged over by $a, b, \ldots$

Definition 4. *The formulae Φ of SL are given by the grammar*

$$\Phi ::= \texttt{true} \mid a \mid \neg\Phi \mid \Phi \wedge \Phi \mid \vec{\rho}\,\Phi[\Phi] \mid \overleftarrow{\rho}\,\Phi[\Phi]$$

We denote the Boolean operators $\texttt{false} = \neg\texttt{true}$ and $(\Phi \vee \Phi) = \neg(\neg\Phi \wedge \neg\Phi)$. We also denote $\vec{\mathcal{N}}\,\Phi = \vec{\rho}\,\Phi[\texttt{false}]$ and $\overleftarrow{\mathcal{N}}\,\Phi = \overleftarrow{\rho}\,\Phi[\texttt{false}]$, which for our models are the equivalent of next and previous in temporal logics (as made precise later).

Let us now consider the semantics. Since we focus on complete spaces, we may equivalently describe our models in terms of relations. Thus, a model $\mathcal{T}$ is a four-tuple $\langle S, R, P, L\rangle$ such that S is a set of points, $R : S \to 2^S$ a relation, P a set of atomic propositions, and $L : P \to 2^S$ a labelling function. We also define the standard notion of spatial path in $\mathcal{T}$ from point s_0 to point s_n, i.e., a sequence $s_0 \ldots s_n$ with $n \geq 1$ such that $s_i \in R(s_{i-1})$ for all $i = 1 \ldots n$.

Definition 5. *Let $\mathcal{T}$ be a model. The semantics of a SL formula Φ with respect to a point $s \in S$ is given by the rules*

- $s \models \texttt{true}$
- $s \models a$ *if* $s \in L(a)$
- $s \models \neg\Phi$ *if* $s \not\models \Phi$
- $s \models \Phi_1 \wedge \Phi_2$ *if* $s \models \Phi_1$ *and* $s \models \Phi_2$
- $s \models \vec{\rho}\,\Phi_1[\Phi_2]$ *if there exists a spatial path* $s s_1 \ldots s_n$ *in* $\mathcal{T}$ *such that* $s_n \models \Phi_1$ *and* $s_j \models \Phi_2$ *for all* $j = 1 \ldots n-1$
- $s \models \overleftarrow{\rho}\,\Phi_1[\Phi_2]$ *if there exists a spatial path* $s_0 \ldots s_{n-1} s$ *in* $\mathcal{T}$ *such that* $s_0 \models \Phi_1$ *and* $s_j \models \Phi_2$ *for all* $j = 1 \ldots n-1$

The derived Boolean operators behave as expected, e.g. $s \not\models \texttt{false}$ for all states s. We recover the intuitive meaning of $\vec{\mathcal{N}}\,\Phi$ (hence, the existence of a direct connection between two points) as $\vec{\rho}\,\Phi[\texttt{false}]$, since $s \models \vec{\rho}\,\Phi[\texttt{false}]$ is equivalent to say that $s_1 \models \Phi$ for some $s_1 \in R(s)$. Similarly for $\overleftarrow{\mathcal{N}}$ with respect to R^{-1}. Finally, note that $\vec{\mathcal{N}}$ and $\overleftarrow{\mathcal{N}}$ distribute over the Boolean disjunction operator, so that e.g. $s \models \vec{\mathcal{N}}\,(\Phi_1 \vee \Phi_2)$ iff $s \models (\vec{\mathcal{N}}\,\Phi_1) \vee (\vec{\mathcal{N}}\,\Phi_2)$.

Lemma 2. *Let $\mathcal{T}$ be a model, $s \in S$ a point, and Φ_1, Φ_2 SL formulae. Then $s \models \vec{\rho}\,\Phi_1[\Phi_2]$ iff $s \models \vec{\mathcal{N}}\,\Phi_1 \vee \vec{\mathcal{N}}\,(\Phi_2 \wedge \vec{\rho}\,\Phi_1[\Phi_2])$ (and similarly for $\overleftarrow{\rho}\,\Phi_1[\Phi_2]$).*

Proof. Let $s \models \vec{\rho}\,\Phi_1[\Phi_2]$. It holds if there exists a path $s s_1 \ldots s_n$ in $\mathcal{T}$ such that $s_n \models \Phi_1$ and $s_j \models \Phi_2$ for all $j = 1 \ldots n-1$. Let us assume that $n = 1$. This is equivalent to say that $s_1 \models \Phi_1$, hence $s \models \vec{\mathcal{N}}\,\Phi_1$. So, let $n > 1$. This means that $s_1 \models \Phi_2$, $s_n \models \Phi_1$, and $s_j \models \Phi_2$ for all $j = 2 \ldots n-1$, which is in turn equivalent to state that $s \models \vec{\mathcal{N}}\,(\Phi_2 \wedge \vec{\rho}\,\Phi_1[\Phi_2])$.

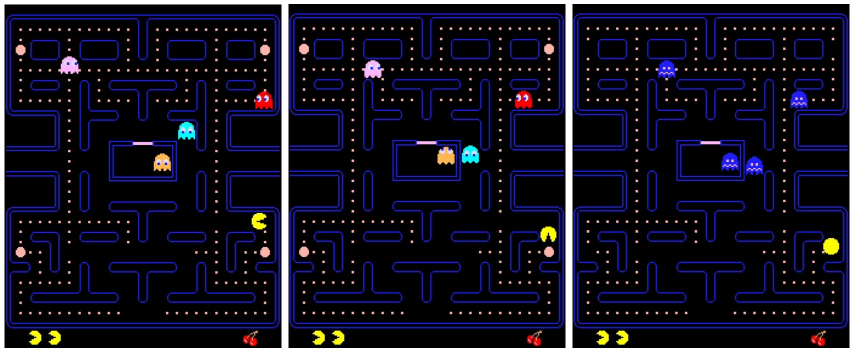

Fig. 1. A sequence of Pac-Man frames: ghosts turn to blue immediately after frame 2.

Example 1. Consider our running example, in particular the first frame of Figure 1. As said above, we assume we have a set of atomic propositions AP denoting colours. There is only one area satisfying the formula *orange*, namely the orange ghost. On the other hand, three different areas satisfy *yellow* and, for the moment being, we are not able to distinguish the active Pac-Man from the ones representing the remaining lives. However, we can already check an interesting property. So, let $ghost = orange \vee pink \vee lightBlue \vee red$. The pixels of a Pac-Man that is going to be caught by a ghost are identified via the formula $yellow \wedge \vec{\rho}\, ghost[yellow]$. Such formula finds all the yellow pixels that are connected, via a path of yellow ones (except the last one, see Definition 5), to a pixel belonging to a ghost. Indeed, no such pixel exists in the three frames considered.

3.1 Denotational semantics of SL

The denotational meaning of a formula Φ is going to be a set of points in our model $\mathcal{T}$. The interpretation of the Boolean and the next and previous step operators is immediate: only the reachability operators need some care.

Definition 6. *Let $\mathcal{T}$ be a model. The denotational semantics of a SL formula Φ is given by the rules*

- $[\![\texttt{true}]\!] = S$
- $[\![a]\!] = L(a)$
- $[\![\neg\Phi]\!] = [\![\Phi]\!]^c = S \setminus [\![\Phi]\!]$
- $[\![\Phi_1 \wedge \Phi_2]\!] = [\![\Phi_1]\!] \cap [\![\Phi_2]\!]$
- $[\![\vec{\mathcal{N}}\,\Phi]\!] = 2^{R^{-1}}([\![\Phi]\!]) = \{s \in S \mid R(s) \cap [\![\Phi]\!] \neq \emptyset\}$
- $[\![\overleftarrow{\mathcal{N}}\,\Phi]\!] = 2^{R}([\![\Phi]\!]) = \{s \in S \mid R^{-1}(s) \cap [\![\Phi]\!] \neq \emptyset\}$
- $[\![\vec{\rho}\,\Phi_1[\Phi_2]]\!] = \texttt{lfp}_Z\,([\![\vec{\mathcal{N}}\,\Phi_1]\!] \cup [\![\vec{\mathcal{N}}\,(\Phi_2 \wedge Z)]\!])$
- $[\![\overleftarrow{\rho}\,\Phi_1[\Phi_2]]\!] = \texttt{lfp}_Z\,([\![\overleftarrow{\mathcal{N}}\,\Phi_1]\!] \cup [\![\overleftarrow{\mathcal{N}}\,(\Phi_2 \wedge Z)]\!])$

The semantics associates a set of points to a formula. The interpretation of the $\vec{\mathcal{N}}$ and $\overleftarrow{\mathcal{N}}$ operators is clearly monotone with respect to subset inclusion, thus the least fix-point in the semantics of the $\vec{\rho}$ and $\overleftarrow{\rho}$ operators are well-defined.

Remark 3. For the sake of simplicity, in Definition 6 we considered $\vec{\mathcal{N}}$ and $\overleftarrow{\mathcal{N}}$ as primitive operators, instead of derived ones. However, it is easy to see that $[\![\vec{\rho}\ \Phi[\mathtt{false}]]\!] = \mathtt{lfp}_Z\,([\![\vec{\mathcal{N}}\,\Phi]\!] \cup ([\![\vec{\mathcal{N}}\,(\mathtt{false} \wedge Z)]\!])) = [\![\vec{\mathcal{N}}\,\Phi]\!]$, and analogously $[\![\overleftarrow{\rho}\ \Phi[\mathtt{false}]]\!] = [\![\overleftarrow{\mathcal{N}}\,\Phi]\!]$. Also note that $[\![\vec{\rho}\ \mathtt{false}[\Phi]]\!] = \mathtt{lfp}_Z\,([\![\vec{\mathcal{N}}\,\mathtt{false}]\!] \cup ([\![\vec{\mathcal{N}}\,(\Phi \wedge Z)]\!])) = \emptyset$, and again analogously $[\![\overleftarrow{\rho}\ \mathtt{false}[\Phi]]\!] = \emptyset$.

Proposition 4. *Let $\mathcal{T}$ be a model, $s \in S$ a point, and Φ a SL formula. Then $s \models \Phi$ iff $s \in [\![\Phi]\!]$.*

Proof. The proof is immediate for all operators except reachability. Consider e.g. the next operator: we have that $s \models \vec{\mathcal{N}}\,\Phi$ iff $s_1 \models \Phi$ for some $s_1 \in R(s)$ iff $R(s) \cap [\![\Phi]\!] \neq \emptyset$, the latter by inductive hypothesis. And we noted in Remark 3 that the semantics of the derived operators is respected, i.e. $[\![\vec{\rho}\,\Phi[\mathtt{false}]]\!] = [\![\vec{\mathcal{N}}\,\Phi]\!]$.

Now, recall that by Lemma 2 $s \models \vec{\rho}\Phi_1[\Phi_2]$ iff $s \models \vec{\mathcal{N}}\,\Phi_1 \vee \vec{\mathcal{N}}\,(\Phi_2 \wedge \vec{\rho}\Phi_1[\Phi_2])$.

($\Longrightarrow$) By induction on the length of the path $ss_1 \ldots s_n$ verifying $s \models \vec{\rho}\Phi_1[\Phi_2]$. If $n = 1$, then $s_1 \models \Phi_1$, hence $s_1 \in [\![\Phi_1]\!]$ and $s \in [\![\vec{\mathcal{N}}\,\Phi_1]\!]$, Otherwise, $s_1 \models \Phi_2 \wedge \vec{\rho}\Phi_1[\Phi_2]$ with a path of length $n-1$, hence $s_1 \in [\![\Phi_2 \wedge \vec{\rho}\Phi_1[\Phi_2]]\!]$ and $s \in [\![\vec{\mathcal{N}}\,(\Phi_2 \wedge \vec{\rho}\Phi_1[\Phi_2]]\!])$. In both cases, we have that $s \in [\![\vec{\rho}\,\Phi_1[\Phi_2]]\!]$.

($\Longleftarrow$) By induction on the number r of recursive steps $Z_1,\ Z_2\ \ldots\ Z_r$. If $r = 1$, then $s \in [\![\vec{\mathcal{N}}\,\Phi_1]\!]$, hence there exists $s_1 \in R(S) \cap [\![\Phi_1]\!]$, thus $s_1 \in R(S)$ and $s_1 \models [\![\Phi_1]\!]$. For $r = n+1$, either $s \in [\![\vec{\mathcal{N}}\,\Phi_1]\!]$, and we fall back to the previous case, or $s \in [\![\vec{\mathcal{N}}\,(\Phi_2 \wedge Z_n)]\!]$. Hence there exists $s_1 \in R(S) \cap [\![\Phi_2]\!] \cap [\![Z_n]\!]$, so the by inductive hypothesis $s_1 \models \Phi_2 \wedge \vec{\rho}\Phi_1[\Phi_2]$. In both cases, we have that $s \models \vec{\rho}\,\Phi_1[\Phi_2]$.

3.2 SL vs. CTL

We make here precise the connection between SL and CTL. The state formulas for the existential fragment of CTL (ECTL) can be expressed by the grammar

$$\Psi ::= \mathtt{true} \mid a \mid \neg\Psi \mid \Psi \wedge \Psi \mid \exists\mathsf{O}\Psi \mid \exists\mathsf{U}(\Psi, \Psi)$$

Note that this fragment is not as expressive as CTL, since it is missing the operators $\forall\mathsf{O}\Psi$ and $\forall\mathsf{U}(\Psi, \Psi)$. And while the former is CTL-equivalent to $\neg\exists\mathsf{O}\neg\Psi$, the latter cannot be expressed in the fragment: it requires the operator $\exists\Box$.

Let us now prove the equivalence of ECTL with the forward fragment of SL (FSL), i.e. SL without the backward operator $\overleftarrow{\rho}$. We do not recall here the semantics for CTL, and we refer the reader to a standard reference such as [3].

The encodings. For any FSL formula Φ we must obtain an ECTL formula $[\![\Phi]\!]$ such that for any model $\mathcal{T}$ and state s in $\mathcal{T}$ we have that $s \models_{SL} \Phi$ iff $s \models_{CTL} [\![\Phi]\!]$. Clearly, the Boolean operators are mapped one-to-one, while $\vec{\rho}\,\Phi_1[\Phi_2]$ is mapped into $\exists\mathsf{O}(\exists\mathsf{U}([\![\Phi_2]\!], [\![\Phi_1]\!]))$. Note that, as a derived operator, $\vec{\mathcal{N}}\,\Phi$ is mapped into $\exists\mathsf{O}(\exists\mathsf{U}(\mathtt{false}, [\![\Phi]\!]))$, which is CTL-equivalent to $\exists\mathsf{O}[\![\Phi]\!]$.

Viceversa, for any ECTL formula Ψ we must obtain a FSL formula $\|\Psi\|$. As before, the Boolean operators are mapped one-to-one, while instead $\exists\mathsf{O}\Psi$ is mapped to $\vec{\mathcal{N}}\,\|\Psi\|$ and $\exists\mathsf{U}(\Psi_1, \Psi_2)$ is mapped to $\|\Psi_2\| \vee (\|\Psi_1\| \wedge \vec{\rho}\,\|\Psi_2\|[\|\Psi_1\|])$. Again, for any model $\mathcal{T}$ and state s in $\mathcal{T}$ we have that $s \models_{CTL} \Psi$ iff $s \models_{SL} \|\Psi\|$.

Encodings are mutually inverse. We proceed by structural induction, assuming that for the sub-formulae it holds that $[\![\|\Psi\|]\!]$ and Ψ are CTL-equivalent and $\|[\![\Phi]\!]\|$ and Φ are SL-equivalent.

Starting from ECTL, we have that

- $[\![\|\exists\mathsf{O}\Psi\|]\!] = [\![\vec{\mathcal{N}}\,\|\Psi\|]\!] = \exists\mathsf{O}(\exists\mathsf{U}(\mathtt{false}, [\![\|\Psi\|]\!]))$
- $[\![\|\exists\mathsf{U}(\Psi_1, \Psi_2)\|]\!] = [\![\|\Psi_2\| \vee (\|\Psi_1\| \wedge \vec{\rho}\;\|\Psi_2\|[\|\Psi_1\|])]\!] = [\![\|\Psi_2\|]\!] \vee ([\![\|\Psi_1\|]\!] \wedge [\![\vec{\rho}\;\|\Psi_2\|[\|\Psi_1\|]]\!]) = [\![\|\Psi_2\|]\!] \vee ([\![\|\Psi_1\|]\!] \wedge \exists\mathsf{O}(\exists\mathsf{U}([\![\|\Psi_1\|]\!], [\![\|\Psi_2\|]\!])))$

and the result follows since for the former case $\exists\mathsf{U}(\mathtt{false}, [\![\|\Psi\|]\!])$ is CTL-equivalent to $[\![\|\Psi\|]\!]$ and for the latter case it is the well-known expansion law for $\exists\mathsf{U}$.

Moving from FSL, we have that

- $\|[\![\vec{\rho}\,\Phi_1[\Phi_2]]\!]\| = \|\exists\mathsf{O}(\exists\mathsf{U}([\![\Phi_2]\!], [\![\Phi_1]\!]))\| = \vec{\mathcal{N}}\,\|\exists\mathsf{U}([\![\Phi_2]\!], [\![\Phi_1]\!])\| = \vec{\mathcal{N}}\,([\![\|\Phi_1\|]\!] \vee ([\![\|\Phi_2\|]\!] \wedge \vec{\rho}\;[\![\|\Phi_1\|]\!][[\![\|\Phi_2\|]\!]])$

The two formulae are SL-equivalent, as shown in Lemma 2.

4 Quantified spatial logics

We now move to a Quantified Spatial Logic (QSL). In the following, we fix a set of typed variables $V = V_P \uplus V_S$ ranged over by $x, y, x_P, y_P, x_S, y_S \ldots$

Definition 7. *The formulae Φ of QSL are given by the grammar*

$$\Phi ::= \mathtt{true} \mid a \mid x \mid x = y \mid \neg\Phi \mid \Phi \wedge \Phi \mid \vec{\rho}\,\Phi[\Phi] \mid \overleftarrow{\rho}\,\Phi[\Phi] \mid \exists_x.\Phi$$

Definition 8. *Let $\mathcal{T}$ be a model. The semantics of a QSL formula Φ with respect to a point $s \in S$ and a substitution $\eta : V \rightharpoonup P \uplus S$ is given by the rules*

- $s, \eta \models x_P$ *if* $s \in L(\eta(x_P))$
- $s, \eta \models x_S$ *if* $s = \eta(x_S)$
- $s, \eta \models x = y$ *if* $\eta(x) = \eta(y)$
- $s, \eta \models \exists_{x_P}.\Phi$ *if there exists a proposition* a_1 *such that* $s, \eta[{}^{a_1}/_{x_P}] \models \Phi$
- $s, \eta \models \exists_{x_S}.\Phi$ *if there exists a point* s_1 *such that* $s, \eta[{}^{s_1}/_{x_S}] \models \Phi$

for $\eta[{}^{a_1}/_{x_P}]$ and $\eta[{}^{s_1}/_{x_S}]$ the standard extensions of a substitution η.

For the sake of readability, we showed only the rules for the variables and the existential operators, and implicitly assumed that equality $x = y$ is well-typed.

Remark 4. Variables may take values either in points or in atomic propositions. Hence, we have statements such as $s, \eta \models x \wedge y$ with $\eta(x)$ a point and $\eta(y)$ an atomic proposition, which still has a clear semantics: it holds if $s = \eta(x)$ and $s \in L(\eta(y))$. As recalled, we implicitly have typed equality $x =_\tau y$ for variables x, y of the same type τ, which is either S for points or P for atomic propositions. With respect to [10], we lack an explicit constant `this` for characterising the current state, which can be obtained by using a point variable x occurring in a formula Φ and simply checking $s \models \exists_x.(x \wedge \Phi)$. In general, the equality $x_S = a$, meaning that the point associated to x_S by a substitution η satisfies proposition a, is recovered as $x_S \wedge a$. Also lacking are equalities $x_P = a$ for proposition a: they seem less relevant, and could be added with little effort.

Remark 5. A further step along the lines above is to assume that variables take values in sets of points, i.e. $\eta : V \to 2^S$, obtaining a second-order quantification. It would simply mean to add an additional type for second-order variables and possibly a monadic operator $\in$, as in $x \in X$. Note that in this case the equality $x = y$ for point variables could be derived as $\forall_X.\ x \in X \iff y \in X$.

4.1 Denotational semantics for QSL

The denotational meaning of a QSL formula Φ is going to be a set of points in our model $\mathcal{T}$. We define our denotational mapping $[\![\cdot]\!]_\eta$ as follows.

Definition 9. *Let $\mathcal{T}$ be a model. The denotational semantics of a QSL formula Φ with respect to a substitution η is given by the rules*

- $[\![x_P]\!]_\eta = L(\eta(x_P))$
- $[\![x_S]\!]_\eta = \{\eta(x_S)\}$
- $[\![x = y]\!]_\eta = \begin{cases} S \text{ if } \eta(x) = \eta(y) \\ \emptyset \text{ otherwise} \end{cases}$
- $[\![\exists_{x_P}.\Phi]\!]_\eta = \bigcup_{a \in P} [\![\Phi]\!]_{\eta[{}^a/_{x_P}]}$
- $[\![\exists_{x_S}.\Phi]\!]_\eta = \bigcup_{s \in S} [\![\Phi]\!]_{\eta[{}^s/_{x_S}]}$

As before, we just showed the rules for variables and existential operators.

Remark 6. It should be no surprise now that the equality $[\![\exists_x.\vec{\mathcal{N}}\Phi]\!]_\eta = [\![\vec{\mathcal{N}}\exists_x.\Phi]\!]_\eta$ holds for any η, and similarly for $\overleftarrow{\mathcal{N}}$. Indeed, the shape of a single frame never changes, hence QSL satisfies what is called the domain-preserving property.

Now, let $\bot : V \rightharpoonup P \uplus S$ denote the always undefined substitution.

Proposition 5. *Let $\mathcal{T}$ be a model, $s \in S$ a point, and Φ a closed QSL formula. Then $s, \bot \models \Phi$ iff $s \in [\![\Phi]\!]_\bot$.*

Remark 7. Quantification over atomic proposition is intended to model the idea of quantifying over "labels" that identify sets of points sharing similar features, in such a way that the number of available labels is infinite and model-dependent. This does not imply that the set of labels that are *present* in each state is infinite: it could as well be that, in a system with infinite states, the number of labels of *each state* is finite, but no state has the same set of labels. In this situation, typical e.g. of nominal computations [23], it might not be possible to know in advance which labels will be present in a state of the model. But this does not rule out the possibility of asking meaningful questions, such as "is there a point labelled with x_P in the current state, which in the next state will *not* be labelled with x_P and *near* to a point labelled with x_P?", which could be interpreted as the entity denoted by x_P has moved by one step in one instant of time.

Although it is perhaps easier to grasp the intuition when models have a temporal aspect, the idea is also useful in purely spatial situations. One case often occurring in computational imaging is that of reasoning about *connected components*. Consider a spatial formula ϕ interpreted over a digital image. No matter what ϕ is, the semantics will identify the set of points S on which ϕ holds. In many situations one could be interested in questions such as "identify the set of points S' that belong to a *connected region* R of S, which also satisfies ψ". In our view, connectedness is not a primitive of the logical language (as connectedness is just one example of application of quantification over atomic propositions!). Rather, the *model* must contain enough information to reason – in this case – about connected components, by having a different atomic proposition for each component[1]. In this situation, one does not know in advance neither how many components (hence, atomic propositions) will be available, nor the exact set of labels, but still, existential quantification over atomic propositions can be used.

Example 2. Using the aforementioned encoding of connected component labels as atomic propositions, we are able to identify entities in a given space. Continuing from Example 1, we now assume that for each frame the set of atomic properties includes colours as well as the labels of the connected components of the yellow pixels. We can now characterise in each frame the pixels on the border of the active Pac-Man as $\Phi = \mathit{yellow} \wedge \forall x_P.\ (\vec{\rho}\ (x_P \wedge \mathit{yellow})[\mathit{black}] \implies x_P)$, since the active Pac-Man cannot reach those outside while these latter are mutually reachable, and the whole active Pac-Man via the formula $\mathit{yellow} \wedge \vec{\rho}\ \Phi[\mathit{yellow}]$.

5 Spatio-temporal logics

The definitions below have the following rationale. In analysing video frames we basically deal with sequences of graphs, each one of them a snapshot of an image. The structure of the graph remains the same: only the labelling changes, i.e the atomic propositions each point satisfies. Also, note that when we state properties of sequences of graphs, we often do not even have a way to generate such sequences. Think e.g. about the scans of the brain: they are given by physicians, and they are not obtained by a set of rules, since they are just snapshots taken at certain intervals of time. We might thus have a single trace as model. This is the reason for the choice of linear time, hence of our Spatio-Temporal Logic (STL): the following proposals could be easily rephrased in terms of computational trees.

Definition 10. *The formulae Φ of STL are given by the grammar*

$$\Phi ::= \texttt{true} \mid a \mid \neg\Phi \mid \Phi \wedge \Phi \mid \vec{\rho}\ \Phi[\Phi] \mid \overleftarrow{\rho}\ \Phi[\Phi] \mid \mathsf{O}\Phi \mid \mathsf{U}(\Phi, \Phi)$$

[1] In model checking, this is accomplished at model definition time, by including a *non-logical* operator which performs a *labelling of connected components*, taking as input a Boolean-labelled frame and returning a integer-labelled frame, where each connected component is identified by a unique integer. See [9] where the on-GPU variant of the spatial model checker VoxLogicA has been endowed with such a primitive.

A spatio-temporal model $\mathcal{S}$ is a four-tuple $\langle S, P, R, \Lambda_0 \rangle$, where S is a set of points, P a set of atomic propositions, $R : S \to 2^S$ a (spatial) relation, $\Lambda_0 \subset \Lambda^+$ a set of temporal traces of length at least 1, for $\Lambda = \{L \mid L : P \to 2^S\}$ the set of labelings. We give the semantics of the formulae with respect to a point s and a finite trace λ. Given a temporal trace $\lambda = L_0 L_1 \dots, L_n$, we denote by $\lambda(i)$ the sequence $L_i L_{i+1} \dots$, by λ_i its i-th component L_i, and with $l(\lambda)$ its length $n+1$.

Definition 11. *Let $\mathcal{T}$ be a spatio-temporal model. The semantics of a STL formula Φ with respect to a point $s \in S$ and a temporal trace $\lambda \in \Lambda_0$ is given by the rules*

- $s, \lambda \models \mathsf{O}\Phi$ *if* $1 < l(\lambda)$ *and* $s, \lambda(1) \models \Phi$
- $s, \lambda \models \mathsf{U}(\Phi_1, \Phi_2)$ *if there exists* $k < l(\lambda)$ *such that* $s, \lambda(k) \models \Phi_2$ *and* $s, \lambda(j) \models \Phi_1$ *for all* $j = 0 \dots k-1$

Remark 8. Since we are using *finite* temporal traces, a few considerations are in order. As a start, a formula $\mathsf{O}\Phi$ is satisfiable by a temporal trace if it is of length at least two, so that $\mathtt{last} = \neg \mathsf{O}\mathtt{true}$ actually characterises its last component. Such an operator allows an easy characterisation for the nesting of temporal operators, since $\Box\Diamond\Phi$ and $\Diamond\Box\Phi$ are equivalent to $\Diamond(\mathtt{last} \wedge \Phi)$ [19].

A related question is which axioms hold. As an example, $\neg\mathsf{O}\Phi$ and $\mathsf{O}\neg\Phi$ are equivalent only for temporal traces of length at least two, since $\mathsf{O}\Phi$ is always false for temporal traces of length 1. Instead, the usual unfolding axiom for the until operator holds, that is, $s, \lambda \models \mathsf{U}(\Phi_1, \Phi_2)$ iff $s, \lambda \models \Phi_2 \vee (\Phi_1 \wedge \mathsf{O}\mathsf{U}(\Phi_1, \Phi_2))$.

The interaction between spatial and temporal operators needs to be explored. For example, $\vec{\rho}\, \mathsf{O}a[\mathsf{O}b]$ is equivalent to $\mathsf{O}(\vec{\rho}\, a[b])$, since the structure of the model (points and their relations) never changes during the steps of a temporal trace.

5.1 Denotational semantics of STL

The denotational meaning of a formula Φ is going to be a set of points in our model $\mathcal{T}$. We define our denotational mapping $[\![\cdot]\!]_\lambda$ as follows.

Definition 12. *Let $\mathcal{T}$ be a spatio-temporal model. The denotational semantics of a STL formula Φ with respect to a temporal trace $\lambda \in \Lambda_0$ is given by the rules*

- $[\![\mathsf{O}\Phi]\!]_\lambda = \begin{cases} [\![\Phi]\!]_{\lambda(1)} & \textit{if } 1 < l(\lambda) \\ \emptyset & \textit{otherwise} \end{cases}$
- $[\![\mathsf{U}(\Phi_1, \Phi_2)]\!]_\lambda = \mathtt{lfp}_W\, ([\![\Phi_2]\!]_\lambda \cup ([\![\Phi_1]\!]_\lambda \cap [\![\mathsf{O}\, W]\!]_\lambda))$

As before, we presented the mapping only for the newly introduced temporal operators. As for the reachability operators, the fix-point for U is well-defined.

Proposition 6. *Let $\mathcal{T}$ be a spatio-temporal model, $s \in S$ a point, $\lambda \in \Lambda_0$ a temporal trace, and Φ a STL formula. Then $s, \lambda \models \Phi$ iff $s \in [\![\Phi]\!]_\lambda$.*

Proof. Similarly to the operators of spatial logics in the proof of Proposition 4, we will basically proceed by induction on the structure of the formulae, considering here also the length of the temporal trace. We just look at the additional temporal operators, noting that it is obvious for the next operator $\mathsf{O}\Phi$. Recall, see Remark 8, that formulae $\mathsf{U}(\Phi_1, \Phi_2)$ and $\Phi_2 \vee (\Phi_1 \wedge \mathsf{OU}(\Phi_1, \Phi_2))$ are equivalent.

($\Longleftarrow$) By induction on the structure of the formulae and the length of the temporal trace. If $s, \lambda \models \Phi_2 \vee (\Phi_1 \wedge \mathsf{OU}(\Phi_1, \Phi_2))$, then either $s, \lambda \models \Phi_2$, hence $s \in [\![\Phi_2]\!]_\lambda$ by inductive hypothesis, or $s, \lambda \models \mathsf{OU}(\Phi_1, \Phi_2)$, thus $s \models \Phi_1$ and $s, \lambda(1) \models \mathsf{U}(\Phi_1, \Phi_2)$, hence $s \in [\![\Phi_1]\!]_\lambda \cap [\![\mathsf{O}\, W]\!]_\lambda$ by inductive hypothesis.

($\Longrightarrow$) By induction on the number r of recursive steps W_1, W_2 ... W_r and the length of the temporal trace. If $r = 1$, then $s \in [\![\Phi_2]\!]_\lambda$, and we are done by inductive hypothesis. For $r = n + 1$, we have that either $s \in [\![\Phi_2]\!]_\lambda$, and we fall back to the previous case, or $s \in [\![\Phi_2]\!] \cap [\![\mathsf{O}W_n]\!]_\lambda$, and in particular $s \in [\![W_n]\!]_{\lambda(1)}$, Thus by inductive hypothesis $s, \lambda \models \Phi_2 \wedge \mathsf{OU}(\Phi_1, \Phi_2))$.

6 All together now

Recall that with our logics we aim to state properties about the single snapshots of a sequence, detailing their changes along time. The Quantified Spatio-Temporal Logic (QSTL) is obtained just by the combination of all the operators introduced so far, thus quantifying "globally" along the whole length of a trace.

Definition 13. *The formulae Φ of QSTL are given by the grammar*

$$\Phi ::= \texttt{true} \mid a \mid x \mid x = y \mid \neg\Phi \mid \Phi \wedge \Phi \mid \vec{\rho}\,\Phi[\Phi] \mid \overleftarrow{\rho}\,\Phi[\Phi] \mid \mathsf{O}\Phi \mid \mathsf{U}(\Phi, \Phi) \mid \exists_x.\Phi$$

Definition 14. *Let $\mathcal{T}$ be a spatio-temporal model. The semantics of a QSTL formula Φ with respect to a point $s \in S$, a substitution $\eta : V \rightharpoonup P \uplus S$, and a temporal trace $\lambda \in \Lambda_0$ is given by the rules*

- $s, \eta, \lambda \models \texttt{true}$
- $s, \eta, \lambda \models a$ *if* $a \in \lambda_0(s)$
- $s, \eta, \lambda \models x_P$ *if* $s \in \lambda_0(\eta(x_P))$
- $s, \eta, \lambda \models x_S$ *if* $s = \eta(x_S)$
- $s, \eta, \lambda \models x = y$ *if* $\eta(x) = \eta(y)$
- $s, \eta, \lambda \models \neg\Phi$ *if* $s, \eta, \lambda \not\models \Phi$
- $s, \eta, \lambda \models \Phi_1 \wedge \Phi_2$ *if* $s, \eta, \lambda \models \Phi_1$ *and* $s, \eta, \lambda \models \Phi_2$
- $s, \eta, \lambda \models \vec{\rho}\ \Phi_1[\Phi_2]$ *if there exists a spatial path* $ss_1 \dots s_n$ *in* $\mathcal{T}$ *such that* $s_n, \eta, \lambda \models \Phi_1$ *and* $s_j, \eta, \lambda \models \Phi_2$ *for all* $j = 1 \dots n-1$
- $s, \eta, \lambda \models \overleftarrow{\rho}\ \Phi_1[\Phi_2]$ *if there exists a spatial path* $s_0 \dots s_{n-1}s$ *in* $\mathcal{T}$ *such that* $s_0, \eta, \lambda \models \Phi_1$ *and* $s_j, \eta, \lambda \models \Phi_2$ *for all* $j = 1 \dots n-1$
- $s, \eta, \lambda \models \mathsf{O}\Phi$ *if* $1 < l(\lambda)$ *and* $s, \eta, \lambda(1) \models \Phi$
- $s, \eta, \lambda \models \mathsf{U}(\Phi_1, \Phi_2)$ *if there exists* $k < l(\lambda)$ *such that* $s, \eta, \lambda(k) \models \Phi_2$ *and* $s, \eta, \lambda(j) \models \Phi_1$ *for all* $j = 0 \dots k-1$
- $s, \eta, \lambda \models \exists_{x_P}.\Phi$ *if there exists a proposition* a_1 *such that* $s, \eta[{}^{a_1}/_x], \lambda \models \Phi$
- $s, \eta, \lambda \models \exists_{x_S}.\Phi$ *if there exists a point* s_1 *such that* $s, \eta[{}^{s_1}/_x], \lambda \models \Phi$

We can now combine the denotational mappings seen before to get $[\![\cdot]\!]_{\eta,\lambda}$, and to finally obtain our concluding result.

Proposition 7. *Let $\mathcal{T}$ be a spatio-temporal model, $s \in S$ a point, $\lambda \in \Lambda_0$ a temporal trace, and Φ a QSTL formula. Then $s, \bot, \lambda \models \Phi$ iff $s \in [\![\Phi]\!]_{\bot,\lambda}$.*

Example 3. We shall now discuss a scenario where all the features of the language are needed. This example is aimed at tracking the identity of objects along the temporal axis. As said in Remark 7, quantifiers on atomic propositions are used to assign labels in order to identify entities, being these points or regions. In Example 2 these labels represent connected components. In this case, instead, we assume that, for each ghost, the spatio-temporal model encodes the identity of each "lifespan" (the time between a character first appears on the screen, and the moment it is caught, or the game finishes) via a unique atomic proposition. In other terms, for each ghost and each lifespan, a separate atomic proposition always identifies all the pixels that the ghost occupies on screen.

We use this idea to define a logic formula ϕ that is true at the pixels of the orange ghost, in the current state, if and only if such ghost will be caught by Pac-Man in a subsequent state. We shall use the derived operator "somewhere" defined as $\mathcal{F}\phi = \vec{\rho}\phi[\texttt{true}]$. We define the formula $\mathit{orange} \wedge \exists x_P . x_P \wedge \mathsf{U}(\mathit{true}, \mathcal{F}(x_P \wedge \mathit{blue} \wedge \vec{\mathcal{N}}\mathit{pacman}))$. Note that, if the formula is true at a point s, then that point is orange, and there is an atomic proposition x_P which holds in s, thus, by construction, it represents the identity of the current ghost. Furthermore, by definition of U, such atomic proposition is still true at *some* point s' of the space, in some future state, with s' in contact with a point of Pac-Man, which entails that the ghost is caught in the same sense of Example 1.

7 Conclusions and future works

We developed a quantified spatio-temporal logic, and showed how this can be used to state spatial properties, possibly involving the identity of individuals, in models that evolve along time. The logic thus represents a significant improvement in expressivity with respect to SLCS [16]. Differently from [10], we adopted linear time operators and an operational semantics based on finite traces. We also introduced a denotational semantics and proved its equivalence with the operational one. Despite its simplicity, the Pac-Man example clarifies the usefulness of the logic in applicative domains such as video stream analysis and lesion tracking in medical imaging.

Concerning future works, we plan to investigate decidability and axiomatisations of the logic. Bisimilarity and minimisation of models can be also of interest, akin to the work for SLCS in [15]. As far as applications are concerned, we will aim at developing a prototype spatial model checker combining temporal and existential operators, and to use it in medical imaging case studies.

References

1. Aiello, M., Pratt-Hartmann, I., Benthem, van, J.: Handbook of Spatial Logics. Springer (2007)
2. Awodey, S., Kishida, K.: Topology and modality: The topological interpretation of first-order modal logic. The Review of Symbolic Logic **1**(2), 146–166 (2008). https://doi.org/10.1017/S1755020308080143
3. Baier, C., Katoen, J.P.: Principles of Model Checking. The MIT Press (2008)
4. Banci Buonamici, F., Belmonte, G., Ciancia, V., Latella, D., Massink, M.: Spatial logics and model checking for medical imaging. Software Tools for Technology Transfer **22**(2), 195–217 (2020)
5. Bartocci, E., Gol, E., Haghighi, I., Belta, C.: A formal methods approach to pattern recognition and synthesis in reaction diffusion networks. IEEE Transactions on Control of Network Systems **5**(1), 308–320 (2016)
6. Belmonte, G., Broccia, G., Ciancia, V., Latella, D., Massink, M.: Feasibility of spatial model checking for nevus segmentation. In: Bliudze, S., Gnesi, S., Plat, N., Semini, L. (eds.) FormaliSE@ICSE 2021. pp. 1–12. IEEE (2021)
7. Belmonte, G., Ciancia, V., Latella, D., Massink, M.: Voxlogica: A spatial model checker for declarative image analysis. In: Vojnar, T., Zhang, L. (eds.) TACAS 2019. LNCS, vol. 11427, pp. 281–298. Springer (2019)
8. Bennett, B., Cohn, A., Wolter, F., Zakharyaschev, M.: Multi-dimensional modal logic as a framework for spatio-temporal reasoning. Applied Intelligence **17** (12 2000). https://doi.org/10.1023/A:1020083231504
9. Bussi, L., Ciancia, V., Gadducci, F.: Towards a spatial model checker on gpu. In: Peters, K., Willemse, T.A.C. (eds.) Formal Techniques for Distributed Objects, Components, and Systems. pp. 188–196. Springer International Publishing, Cham (2021)
10. Bussi, L., Ciancia, V., Gadducci, F., Latella, D., Massink, M.: On binding in the spatial logics for closure spaces. In: Margaria, T., Steffen, B. (eds.) ISoLA 2022. LNCS, vol. 13701. Springer (2022)
11. Bussi, L., Ciancia, V., Gadducci, F., Latella, D., Massink, M.: Towards model checking video streams using VoxLogicA on GPUs. In: Bowles, J., Broccia, G., Pellungrini, R. (eds.) DataMod 2021. LNCS, vol. 13268, pp. 78–90. Springer (2022)
12. Cardelli, L., Gordon, A.D.: Anytime, anywhere: Modal logics for mobile ambients. p. 365–377. POPL '00, Association for Computing Machinery, New York, NY, USA (2000). https://doi.org/https://doi.org/10.1145/325694.325742
13. Ciancia, V., Latella, D., Massink, M., Paškauskas, R., Vandin, A.: A tool-chain for statistical spatio-temporal model checking of bike sharing systems. In: Margaria, T., Steffen, B. (eds.) ISoLA 2016. LNCS, vol. 9952, pp. 657–673 (2016)
14. Ciancia, V., Gilmore, S., Grilletti, G., Latella, D., Loreti, M., Massink, M.: Spatio-temporal model checking of vehicular movement in public transport systems. Software Tools for Technology Transfer **20**(3), 289–311 (2018)
15. Ciancia, V., Groote, J.F., Latella, D., Massink, M., de Vink, E.P.: Minimisation of spatial models using branching bisimilarity. In: Chechik, M., Katoen, J., Leucker, M. (eds.) FM 2023. LNCS, vol. 14000, pp. 263–281. Springer (2023)
16. Ciancia, V., Latella, D., Loreti, M., Massink, M.: Specifying and verifying properties of space. In: Díaz, J., Lanese, I., Sangiorgi, D. (eds.) TCS 2014. LNCS, vol. 8705, pp. 222–235. Springer (2014)
17. Gabelaia, D., Kontchakov, R., Kurucz, A., Wolter, F., Zakharyaschev, M.: Combining spatial and temporal logics: Expressiveness vs. complexity. Journal of Artificial Intelligence Research **23**, 167–243 (2005)

18. Gadducci, F., Lluch-Lafuente, A., Vandin, A.: Counterpart semantics for a second-order μ-calculus. Fundamenta Informaticae **118**(1-2), 177–205 (2012)
19. Giacomo, G.D., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: Rossi, F. (ed.) IJCAI 2013. pp. 854–860. IJCAI/AAAI (2013)
20. Hilken, B.P., Rydeheard, D.E.: A first order modal logic and its sheaf models (2001)
21. Kishida, K.: Neighborhood-sheaf semantics for first-order modal logic. Electronic Notes in Theoretical Computer Science **278**, 129–143 (2011). https://doi.org/https://doi.org/10.1016/j.entcs.2011.10.011
22. Nenzi, L., Bartocci, E., Bortolussi, L., Loreti, M.: A logic for monitoring dynamic networks of spatially-distributed cyber-physical systems. Logical Methods in Computer Science **18**(1) (2022)
23. Pitts, A.M.: Nominal Sets: Names and Symmetry in Computer Science. Cambridge Tracts in Theoretical Computer Science, Cambridge University Press (2013). https://doi.org/10.1017/CBO9781139084673
24. Tsigkanos, C., Kehrer, T., Ghezzi, C.: Modeling and verification of evolving cyber-physical spaces. In: Bodden, E., Schäfer, W., van Deursen, A., Zisman, A. (eds.) ESEC/FSE 2017. pp. 38–48. ACM (2017)

A Some hints from quantified modal algebras

This appendix recalls basic notions of (quantified) modal and conjugate algebras, which inspired the way we provided our logics with a denotational semantics.

A.1 Boolean and modal algebra

We recall the basics of boolean and modal algebras and discuss some axioms.

Definition 15. *A Boolean algebra $\mathcal{A}$ is a 6-tuple $\langle A, \vee, 0, \wedge, 1, \neg\rangle$ such that the triples $\langle A, \vee, 0\rangle$ and $\langle A, \wedge, 1\rangle$ are ACI (associative, commutative and with identity) monoids satisfying the usual distributivity and negation rules.*

The usual negation rule means that $a \vee \neg a = 1$ and $a \wedge \neg a = 0$. A Boolean algebra is equivalently described as a *complemented distributive lattice*. In particular $a \vee b = a$ iff $a \wedge b = b$ and $a \leq b$ iff $\neg b \leq \neg a$. The partial order on A is induced by $a \leq b$ if $a \vee b = b$, so that 0 is bottom and 1 is top. A well-known example of such a structure is the boolean algebra of powersets of a set, that gives rise to the algebra $\mathcal{A} = \langle \mathcal{P}(A), \cup, \emptyset, \cap, A, ^c\rangle$. We say that a Boolean algebra $\mathcal{A}$ is *complete* if every subset of A has a least upper bound (LUB).

Definition 16. *A modal algebra $\mathcal{M}$ is a 7-tuple $\langle A, \vee, 0, \wedge, 1, \neg, \Diamond\rangle$ such that the 6-tuple $\langle A, \vee, 0, \wedge, 1, \neg\rangle$ is a Boolean algebra and $\Diamond : A \to A$ is a function satisfying $\Diamond 0 = 0$ and $\Diamond(a \vee b) = \Diamond a \vee \Diamond b$.*

A modal algebra is complete *if the underlying Boolean algebra is complete and $\Diamond(\bigvee_i a_i) = \bigvee_i \Diamond a_i$ for any i.*

Monotonicity of $\Diamond$ is implied by the second axiom, which also yields that $\Diamond 1 = 1$. If $\mathcal{M}$ is finite (i.e. the set A is finite), then $\mathcal{M}$ is obviously complete.

We define the usual derived operator $\Box a = \neg\Diamond\neg a$. Note that $\Box 1 = 1$, $\Box a \wedge b = \Box a \wedge \Box b$, and $\Box$ is monotone with respect to the induced partial order

Remark 9. Modal algebras provide denotational models for propositional modal logics. Assuming a semantical function $[\cdot]$ mapping a formula into an element of the modal algebra chosen as model, the formula ϕ is valid in the logics if $[\phi] = 1$. Also, note that $[\phi \implies \rho] = 1$ is equivalent to prove that $[\phi] \leq [\rho]$, assuming that $[\cdot]$ preserves the operators $\neg$ and $\vee$ (hence, all the operators).

It is immediate that the axiom K, i.e. $\Box(\phi \implies \rho) \implies (\Box\phi \implies \Box\rho)$, holds in any modal algebra. By Boolean manipulation the formula is equivalent to $(\Box\phi \wedge (\Box(\phi \implies \rho)) \implies \Box\rho$. Hence, it suffices to prove that in a modal algebra it holds $(\Box a \wedge \Box(a \implies b)) \leq \Box b$. Due to the distributivity of $\Box$, this is equivalent to prove that $\Box(a \wedge b) \leq \Box b$, which holds by monotonicity.

Also, note that what is called the necessitation rule for modal logics based on K holds, since $a = 1$ implies $\Box a = \Box 1 = 1$.

Definition 17. *Let $\mathcal{M}$ be a modal algebra whose partial order is $\leq$. Its necessity and iteration axioms are $M = a \leq \Diamond a$, $4 = \Diamond\Diamond a \leq \Diamond a$, and $B = a \leq \Box\Diamond a$.*

Axioms are given in terms of the $\Diamond$ operator, but they can be rewritten using the $\Box$ operator, with the reversed inequality. Hence, M and 4 can be equivalently expressed in terms of $\Box$ as $\Box a \leq a$ and $\Box a \leq \Box\Box a$, respectively, as well as B is equivalent to $\Diamond\Box a \leq a$. Note that assuming M and 4 implies that $\Diamond\Diamond a = \Diamond a$.

Remark 10. Axioms M, 4, and B are known as reflexivity, transitivity, and symmetry axioms, respectively, since for modal algebras arising from Kripke frames those are the properties imposed on the underlying relation [**?**]. Modal algebras satisfying M and 4 are called closure algebras and are models of $S4$, while those satisfying all three axioms are called monadic algebras and are models of $S5$.

A.2 Quantified modal algebras

While modal algebras represent models for propositional modal logics, moving to first order quantification require the introduction of *cylindric operators*, a well-known abstraction for existential quantifiers [**?**].

Cylindric operators. We fix a Boolean algebra $\mathcal{A}$ and a set of variables V.

Definition 18 (cylindric Boolean algebras). *A cylindric operator $\exists$ over $\mathcal{A}$ and V is a family of monotone operators $\exists_x : A \to A$ indexed by elements in V such that for all $a, b \in A$ and $x, y \in V$ it holds $a \leq \exists_x a$, $\exists_x \exists y a = \exists_y \exists_x a$, and $\exists_x(a \wedge \exists_x b) = \exists_x a \wedge \exists_x b$.*

Let $a \in A$. The support *of a is the set of variables $sv(a) = \{x \mid \exists_x a \neq a\}$.*

An element of the algebra stands for a formula possibly containing free variables. We restrict our attention to elements a with finite support, i.e., such that $sv(a)$ is finite: this means that a is a formula containing a finite set of variables.

Now we fix a modal algebra $\mathcal{M}$ with underlying Boolean algebra $\mathcal{A}$.

Definition 19 (cylindric modal algebras). *A cylindric operator $\exists$ over $\mathcal{M}$ and V is a cylindric operator over $\mathcal{A}$ and V such that for all $a \in A$ and $x \in V$ it holds $\exists_x \Diamond a = \Diamond \exists_x a$.*

Remark 11. The inequalities $\exists_x \Diamond a \geq \Diamond \exists_x a$ and $\exists_x \Diamond a \leq \Diamond \exists_x a$ are known as *Barcan formula* and *converse Barcan formula* in the literature [**?**]. The axiom in Definition 19 is thus only one of the possible choices, and it boils down to require what is called "domain preservation", namely, the domain is preserved along the evolution. Instead, $\exists_x \Diamond a \leq \Diamond \exists_x a$ witnesses a possible domain restriction, while analogously we may have a domain increase with the reverse $\exists_x \Diamond a \geq \Diamond \exists_x a$.

The axiom implies $sv(\Diamond a) \subseteq sv(a)$, since $\exists_x a = a$ implies $\exists_x \Diamond a = \Diamond \exists_x a = \Diamond a$.

Soft modal algebras. We now show how to build a modal algebra that admits cylindric operators. Let us fix a modal algebra $\mathcal{M}$ with underlying Boolean algebra $\mathcal{A}$ and a set of variables V.

Proposition 8. *Let D be a set of elements, F the set of functions $\eta : V \to D$, and Γ the set of functions $\gamma : F \to A$. The 7-tuple $\mathcal{F} = \langle \Gamma, \vee, 0, \wedge, 1, \neg, \Diamond \rangle$ is a modal algebra, whose operators and constants are lifted from $\mathcal{M}$. If $\mathcal{M}$ is complete, so is $\mathcal{F}$.*

For example, 0 in $\mathcal{F}$ is the function such that $0(\eta) = 0$ for all η, and so on. In particular, note that $\gamma_1 \leq \gamma_2$ means that $\gamma_1(\eta) \leq \gamma_2(\eta)$ for all η.

Let us now additionally fix a set D, and given $\eta : V \to D$, we denote as $\eta[^d/_x]$ the function coinciding with η except for x, where $\eta[^d/_x](x) = d$.

Proposition 9. *Let D be finite. The cylindric operator $\exists$ over $\mathcal{F}$ and V is defined as $(\exists_x \gamma)(\eta) = \bigvee_{d \in D} \gamma(\eta[^d/_x])$.*

If $\mathcal{M}$ is complete, the finiteness of D can be dropped.

Remark 12. By definition, $\exists_x \gamma = \gamma$ means that for all η we have $\bigvee_{d \in D} \gamma(\eta[^d/_x]) = \gamma(\eta)$, which is equivalent to say that for all d we have $\gamma(\eta[^d/_x]) = \gamma(\eta)$. Intuitively, if γ represents a formula possibly containing free variables, x cannot be among them. Conversely, $x \in sv(\gamma)$ if there is a function η and elements $b, c \in D$ such that $\gamma(\eta[^b/_x]) \neq \gamma(\eta[^c/_x])$, intuitively meaning that x does occur free in γ.

A.3 Conjugate modal algebras

Algebras that employ more than one modal operator are said to be *multimodal*. We focus here on a particular kind of such algebras, called *conjugate algebras*.

Definition 20. *A conjugate algebra $\mathcal{D}$ is a 8-tuple $\langle A, \vee, 0, \wedge, 1, \neg, \Diamond_1, \Diamond_2 \rangle$ such that both 7-tuples $\langle A, \vee, 0, \wedge, 1, \neg, \Diamond_1 \rangle$ and $\langle A, \vee, 0, \wedge, 1, \neg, \Diamond_2 \rangle$ are modal algebras and moreover it holds $a \leq \Box_1 \Diamond_2 a \wedge \Box_2 \Diamond_1 a$.*

A conjugate algebra is complete *if both the underlying modal algebras are so.*

What is noteworthy is a well-known characterisation via just the $\Diamond$ operators.

Lemma 3. *$\mathcal{D}$ is a conjugate algebra iff it holds $\Diamond_1 a \wedge b = 0 \Leftrightarrow a \wedge \Diamond_2 b = 0$.*

Remark 13. The lemma is stated by using the more standard notion of the axiom on $\Diamond$. An alternative, friendlier version is $\Diamond_1 a \leq b \Leftrightarrow a \leq \Box_2 b$. The proof of the equivalence between the two axioms is straightforward, and it exploits the following law holding in Boolean algebras, namely $a \wedge b = 0$ iff $a \leq \neg b$.