# Specifying and Verifying Properties of Space

Vincenzo Ciancia, Diego Latella, Michele Loreti, Mieke Massink

# Specifying and Verifying Properties of Space*

Vincenzo Ciancia[1], Diego Latella[1], Michele Loreti[2], and Mieke Massink[1]

[1] Istituto di Scienza e Tecnologie dell'Informazione 'A. Faedo', CNR, Italy
[2] Università di Firenze, Italy

**Abstract.** The interplay between process behaviour and spatial aspects of computation has become more and more relevant in Computer Science, especially in the field of *collective adaptive systems*, but also, more generally, when dealing with systems distributed in physical space. Traditional verification techniques are well suited to analyse the temporal evolution of programs; properties of space are typically not explicitly taken into account. We propose a methodology to verify properties depending upon physical space. We define an appropriate logic, stemming from the tradition of topological interpretations of modal logics, dating back to earlier logicians such as Tarski, where modalities describe neighbourhood. We lift the topological definitions to a more general setting, also encompassing discrete, graph-based structures. We further extend the framework with a spatial *until* operator, and define an efficient model checking procedure, implemented in a proof-of-concept tool.

## 1 Introduction

Much attention has been devoted in Computer Science to formal verification of process behaviour. Several techniques, such as *run-time monitoring* and *model-checking*, are based on a formal understanding of system requirements through *modal* logics. Such logics typically have a *temporal* flavour, describing the flow of events along time, and are interpreted in various kinds of transition structures.

Recently, aspects of computation related to the distribution of systems in physical space have become more relevant. An example is provided by so called *collective adaptive systems*[3], typically composed of a large number of interacting objects. Their global behaviour critically depends on interactions which are often local in nature. Locality immediately poses issues of spatial distribution of objects. Abstraction from spatial distribution may sometimes provide insights in the system behaviour, but this is not always the case. For example, consider a bike (or car) sharing system

* Research partially funded by EU ASCENS (nr. 257414), EU QUANTICOL (nr. 600708), IT MIUR CINA and PAR FAS 2007-2013 Regione Toscana TRACE-IT.

[3] See e.g. the web site of the QUANTICOL project: `http://www.quanticol.eu`

having several parking stations, and featuring twice as many parking slots as there are vehicles in the system. Ignoring the spatial dimension, on average, the probability to find completely full or empty parking stations at an arbitrary station is very low; however, this kind of analysis may be misleading, as in practice some stations are much more popular than others, often depending on nearby points of interest. This leads to quite different probabilities to find stations completely full or empty, depending on the spatial properties of the examined location. In such situations, it is important to be able to predicate over spatial aspects, and eventually find methods to certify that a given formal model of space satisfies specific requirements in this respect. In Logics, there is quite an amount of literature focused on so called *spatial* logics, that is, a spatial interpretation of modal logics. Dating back to early logicians such as Tarski, modalities may be interpreted using the concept of *neighbourhood* in a topological space. The field of spatial logics is well developed in terms of descriptive languages and computability/complexity aspects. However, the frontier of current research does not yet address verification problems, and in particular, discrete models are still a relatively unexplored field.

In this paper, we extend the topological semantics of modal logics to *closure spaces*. As we shall discuss in the paper, this choice is motivated by the need to use non-idempotent *closure operators*. A closure space (also called *Čech closure space* or *preclosure space* in the literature), is a generalisation of a standard topological space, where idempotence of closure is not required. By this, graphs and topological spaces are treated uniformly, letting the topological and graph-theoretical notions of neighbourhood coincide. We also provide a spatial interpretation of the *until* operator, which is fundamental in the classical temporal setting, arriving at the definition of a logic which is able to describe unbounded areas of space. Intuitively, the spatial until operator describes a situation in which it is not possible to "escape" an area of points satisfying a certain property, unless by passing through at least one point that satisfies another given formula. To formalise this intuition, we provide a characterising theorem that relates infinite paths in a closure space and until formulas. We introduce a model-checking procedure that is *linear* in the size of the considered space. A prototype implementation of a spatial model-checker has been made available; the tool is able to interpret spatial logics on digital images, providing graphical understanding of the meaning of formulas, and an immediate form of counterexample visualisation.[4]

[4] Due to lack of space all the proofs are omitted and can be found in [10].

*Related work.* We use the terminology *spatial logics* in the "topological" sense; the reader should be warned that in Computer Science literature, spatial logics typically describe situations in which modal operators are interpreted syntactically, against the structure of agents in a process calculus (see [8,6] for some classical examples). The object of discussion in this research line are operators that quantify e.g., over the parallel subcomponents of a system, or the hidden resources of an agent. Furthermore, logics for graphs have been studied in the context of databases and process calculi (see [7,15], and references), even though the relationship with physical space is often not made explicit, if considered at all. The influence of space on agents interaction is also considered in the literature on process calculi using *named locations* [11].

Variants of spatial logics have also been proposed for the symbolic representation of the contents of images, and, combined with temporal logics, for sequences of images [12]. The approach is based on a discretisation of the space of the images in rectangular regions and the orthogonal projection of objects and regions onto Cartesian coordinate axes such that their possible intersections can be analysed from different perspectives. It involves two spatial until operators defined on such projections considering spatial shifts of regions along the positive, respectively negative, direction of the coordinate axes and it is very different from the topological spatial logic approach.

A successful attempt to bring topology and digital imaging together is represented by the field of *digital topology* [22,25]. In spite of its name, this area studies digital images using models inspired by topological spaces, but neither generalising nor specialising these structures. Rather recently, closure spaces have been proposed as an alternative foundation of digital imaging by various authors, especially Smyth and Webster [23] and Galton [17]; we continue that research line, enhancing it with a logical perspective. Kovalevsky [19] studied alternative axioms for topological spaces in order to recover well-behaved notions of neighbourhood. In the terminology of closure spaces, the outcome is that one may impose closure operators on top of a topology, that do not coincide with topological closure. The idea of interpreting the until operator in a topological space is briefly discussed in the work by Aiello and van Benthem [1,24]. We start from their definition, discuss its limitations, and provide a more fine-grained operator, which is interpreted in closure spaces, and has therefore also an interpretation in topological spaces. In the specific setting of complex and collective adaptive systems, techniques for efficient approximation have been developed in the form of mean-field / fluid-flow analysis (see [5] for

a tutorial introduction). Recently (see e.g., [9]), the importance of spatial aspects has been recognised and studied in this context. In this work, we aim at paving the way for the inclusion of spatial logics, and their verification procedures, in the framework of mean-field and fluid-flow analysis of collective adaptive systems.

## 2 Closure spaces

In this work, we use *closure spaces* to define basic concepts of *space*. Below, we recall several definitions, most of which are explained in [17].

**Definition 1.** *A* closure space *is a pair* $(X, \mathcal{C})$ *where* $X$ *is a set, and the* closure operator $\mathcal{C} : 2^X \to 2^X$ *assigns to each subset of* $X$ *its* closure*, obeying to the following laws, for all* $A, B \subseteq X$*:*

1. $\mathcal{C}(\emptyset) = \emptyset$*;*
2. $A \subseteq \mathcal{C}(A)$*;*
3. $\mathcal{C}(A \cup B) = \mathcal{C}(A) \cup \mathcal{C}(B)$*.*

As a matter of notation, in the following, for $(X, \mathcal{C})$ a closure space, and $A \subseteq X$, we let $\overline{A} = X \setminus A$ be the complement of $A$ in $X$.

**Definition 2.** *Let* $(X, \mathcal{C})$ *be a closure space, for each* $A \subseteq X$*:*

1. *the* interior $\mathcal{I}(A)$ *of* $A$ *is the set* $\overline{\mathcal{C}(\overline{A})}$*;*
2. $A$ *is a* neighbourhood *of* $x \in X$ *if and only if* $x \in \mathcal{I}(A)$*;*
3. $A$ *is* closed *if* $A = \mathcal{C}(A)$ *while it is* open *if* $A = \mathcal{I}(A)$*.*

**Lemma 1.** *Let* $(X, \mathcal{C})$ *be a closure space, the following properties hold:*

1. $A \subseteq X$ *is open if and only if* $\overline{A}$ *is closed;*
2. *closure and interior are monotone operators over the inclusion order, that is:* $A \subseteq B \implies \mathcal{C}(A) \subseteq \mathcal{C}(B)$ *and* $\mathcal{I}(A) \subseteq \mathcal{I}(B)$
3. *Finite intersections and arbitrary unions of open sets are open.*

Closure spaces are a generalisation of *topological spaces*. The axioms defining a closure space are also part of the definition of a *Kuratowski closure space*, which is one of the possible alternative definitions of a topological space. More precisely, a closure space is Kuratowski, therefore a topological space, whenever closure is *idempotent*, that is, $\mathcal{C}(\mathcal{C}(A)) = \mathcal{C}(A)$. We omit the details for space reasons (see e.g., [17] for more information).

Next, we introduce the topological notion of *boundary*, which also applies to closure spaces, and two of its variants, namely the *interior* and *closure* boundary (the latter is sometimes called *frontier*).

**Definition 3.** *In a closure space* $(X, \mathcal{C})$*, the* boundary *of* $A \subseteq X$ *is defined as* $\mathcal{B}(A) = \mathcal{C}(A) \setminus \mathcal{I}(A)$*. The* interior boundary *is* $\mathcal{B}^-(A) = A \setminus \mathcal{I}(A)$*, and the* closure boundary *is* $\mathcal{B}^+(A) = \mathcal{C}(A) \setminus A$*.*

**Proposition 1.** *The following equations hold in a closure space:*

$$\mathcal{B}(A) = \mathcal{B}^+(A) \cup \mathcal{B}^-(A) \tag{1}$$
$$\mathcal{B}^+(A) \cap \mathcal{B}^-(A) = \emptyset \tag{2}$$
$$\mathcal{B}(A) = \mathcal{B}(\overline{A}) \tag{3}$$
$$\mathcal{B}^+(A) = \mathcal{B}^-(\overline{A}) \tag{4}$$
$$\mathcal{B}^+(A) = \mathcal{B}(A) \cap \overline{A} \tag{5}$$
$$\mathcal{B}^-(A) = \mathcal{B}(A) \cap A \tag{6}$$
$$\mathcal{B}(A) = \mathcal{C}(A) \cap \mathcal{C}(\overline{A}) \tag{7}$$

## 3 Quasi-discrete closure spaces

In this section we see how a closure space may be derived starting from a *binary relation*, that is, a *graph*. The following comes from [17].

**Definition 4.** *Consider a set* $X$ *and a relation* $R \subseteq X \times X$*. A closure operator is obtained from* $R$ *as* $\mathcal{C}_R(A) = A \cup \{x \in X \mid \exists a \in A.(a, x) \in R\}$*.*

*Remark 1.* One could also change Definition 4 so that $\mathcal{C}_R(A) = A \cup \{x \in X \mid \exists a \in A.(x, a) \in R\}$, which actually is the definition of [17]. This does not affect the theory presented in the paper. Indeed, one obtains the same results by replacing $R$ with $R^{-1}$ in statements of theorems that explicitly use $R$, and are not invariant under such change. By our choice, closure represents the "least possible enlargement" of a set of nodes.

**Proposition 2.** *The pair* $(X, \mathcal{C}_R)$ *is a closure space.*

Closure operators obtained by Definition 4 are not necessarily idempotent. Lemma 11 in [17] provides a necessary and sufficient condition, that we rephrase below. We let $R^=$ denote the reflexive closure of $R$ (that is, the least relation that includes $R$ and is reflexive).

**Lemma 2.** $\mathcal{C}_R$ *is idempotent if and only if* $R^=$ *is transitive.*

Note that, when $R$ is transitive, so is $R^=$, thus $\mathcal{C}_R$ is idempotent. The vice-versa is not true, e.g., when $(x, y) \in R$, $(y, x) \in R$, but $(x, x) \notin R$.

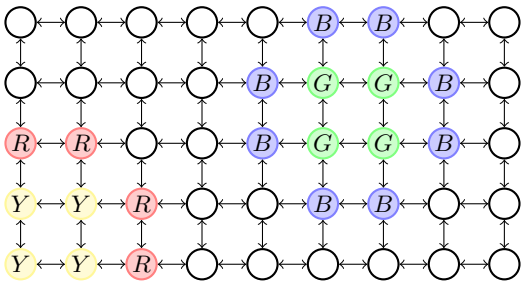


**Fig. 1.** A graph inducing a *quasi-discrete* closure space

*Remark 2.* In topology, open sets play a fundamental role. However, the situation is different in closure spaces derived from a relation $R$. For example, in the case of a closure space derived from a connected symmetric relation, the only open sets are the whole space, and the empty set.

**Proposition 3.** *Given $R \subseteq X \times X$, in the space $(X, \mathcal{C}_R)$, we have:*

$$\mathcal{I}(A) = \{x \in A \mid \neg\exists a \in \overline{A}.(a, x) \in R\} \tag{8}$$

$$\mathcal{B}^{-}(A) = \{x \in A \mid \exists a \in \overline{A}.(a, x) \in R\} \tag{9}$$

$$\mathcal{B}^{+}(A) = \{x \in \overline{A} \mid \exists a \in A.(a, x) \in R\} \tag{10}$$

We note in passing that [16] provides an alternative definition of boundaries for closure spaces obtained from Definition 4, and proves that it coincides with the topological definition (our Definition 3). Closure spaces derived from a relation can be characterised as *quasi-discrete* spaces (see also Lemma 9 of [17] and the subsequent statements).

**Definition 5.** *A closure space is* quasi-discrete *if and only if one of the following equivalent conditions holds: i) each $x \in X$ has a* minimal neighbourhood[5] *$N_x$; ii) for each $A \subseteq X$, $\mathcal{C}(A) = \bigcup_{a \in A} \mathcal{C}(\{a\})$.*

**Theorem 1.** *(Theorem 1 in [17]) A closure space $(X, \mathcal{C})$ is quasi-discrete if and only if there is a relation $R \subseteq X \times X$ such that $\mathcal{C} = \mathcal{C}_R$.*

*Example 1.* Every graph induces a *quasi-discrete* closure space. For instance, we can consider the (undirected) graph depicted in Figure 1. Let $R$ be the (symmetric) binary relation induced by the graph edges, and let

[5] A *minimal neighbourhood* of $x$ is a set that is a neighbourhood of $x$ (Definition 2 (2)) and is included in all other neighbourhoods of $x$.

$Y$ and $G$ denote the set of *yellow* and *green* nodes, respectively. The closure $\mathcal{C}_R(Y)$ consists of all *yellow* and *red* nodes, while the closure $\mathcal{C}_R(G)$ contains all *green* and *blue* nodes. The interior $\mathcal{I}(Y)$ of $Y$ contains a single node, i.e. the one located at the bottom-left in Figure 1. On the contrary, the *interior* $\mathcal{I}(G)$ of $G$ is empty. Indeed, we have that $\mathcal{B}(G) = \mathcal{C}(G)$, while $\mathcal{B}^-(G) = G$ and $\mathcal{B}^+(G)$ consists of the *blue* nodes.

## 4 A Spatial Logic for Closure Spaces

In this section we present a spatial logic that can be used to express properties of closure spaces. The logic features two *spatial operators*: a "one step" modality, turning closure into a logical operator, and a binary *until* operator, which is interpreted spatially. Before introducing the complete framework, we first discuss the design of an *until operator* $\phi \mathcal{U} \psi$.

The spatial logical operator $\mathcal{U}$ is interpreted on points of a closure space. The basic idea is that point $x$ satisfies $\phi \mathcal{U} \psi$ whenever it is included in an area $A$ satisfying $\phi$, and there is "no way out" from $A$ unless passing through an area $B$ that satisfies $\psi$. For instance, if we consider the model of Figure 1, *yellow* nodes satisfy *yellow* $\mathcal{U}$ *red* while *green* nodes satisfy *green* $\mathcal{U}$ *blue*. To turn this intuition into a mathematical definition, one should clarify the meaning of the words *area*, *included*, *passing*, in the context of closure spaces.

In order to formally define our logic, and the *until* operator in particular, we first need to introduce the notion of *model*, providing a context of evaluation for the satisfaction relation, as in $\mathcal{M}, x \models \phi \mathcal{U} \psi$. From now on, fix a (finite or countable) set $P$ of *proposition letters.*

**Definition 6.** *A* closure model *is a pair* $\mathcal{M} = ((X, \mathcal{C}), \mathcal{V})$ *consisting of a closure space* $(X, \mathcal{C})$ *and a valuation* $\mathcal{V} : P \to 2^X$*, assigning to each proposition letter the set of points where the proposition holds.*

When $(X, \mathcal{C})$ is a topological space (that is, $\mathcal{C}$ is idempotent), we call $\mathcal{M}$ a *topological model*, in line with [24], and [1], where the *topological until* operator is presented. We recall it below.

**Definition 7.** *The* topological until *operator* $\mathcal{U}_T$ *is interpreted in a topological model* $\mathcal{M}$ *as* $\mathcal{M}, x \models \phi \mathcal{U}_T \psi \iff \exists A$ *open* $. x \in A \wedge \forall y \in A. \mathcal{M}, y \models \phi \wedge \forall z \in \mathcal{B}(A). \mathcal{M}, z \models \psi$.

The intuition behind this definition is that one seeks for an area $A$ (which, topologically speaking, could sensibly be an open set) where $\phi$

holds, and that is completely surrounded by points where $\psi$ holds. Unfortunately, Definition 7 cannot be translated directly to closure spaces, even if all the used topological notions have a counterpart in the more general setting of closure spaces. Open sets in closure spaces are often too coarse (see Remark 2). For this reason, we can modify Definition 7 by not requiring $A$ to be an *open* set. However, the usage of $\mathcal{B}$ in Definition 7 is not satisfactory either. By Proposition 1 we have $\mathcal{B}(A) = \mathcal{B}^+(A) \cup \mathcal{B}^-(A)$, where $\mathcal{B}^-(A)$ is included in $A$ while $\mathcal{B}^+(A)$ is in $\overline{A}$. For instance, when $\mathcal{B}$ is used in Definition 7, we have that the *green* nodes in Figure 1 do not satisfy $green\ \mathcal{U}_T\ blue$. Indeed, as we remarked in Example 1, the *boundary* of the set $G$ of green nodes coincide with the closure of $G$ that contains both *green* and *blue* nodes.

A more satisfactory definition can be obtained by letting $\mathcal{B}^+$ play the same role as $\mathcal{B}$ in Definition 7 and not requiring $A$ to be an open set. We shall in fact require that $\phi$ is satisfied by *all* the points of $A$, and that in $\mathcal{B}^+(A)$, $\psi$ holds. This allows us to ensure that there are no "gaps" between the region satisfying $\phi$ and that satisfying $\psi$.

### 4.1 Syntax and Semantics of *SLCS*

We can now define $SLCS$: a *Spatial Logic for Closure Spaces*. The logic features boolean operators, a "one step" modality, turning closure into a logical operator, and a spatially interpreted *until* operator. More precisely, as we shall see, the $SLCS$ formula $\phi \mathcal{U} \psi$ requires $\phi$ to hold at least on one point. The operator is similar to a *weak until* in temporal logics terminology, as there may be no point satisfying $\psi$, if $\phi$ holds everywhere.

**Definition 8.** *The syntax of $SLCS$ is defined by the following grammar, where $p$ ranges over $P$:*

$$\Phi ::= p \mid \top \mid \neg\Phi \mid \Phi \wedge \Phi \mid \Diamond\Phi \mid \Phi \mathcal{U} \Phi$$

Here, $\top$ denotes *true*, $\neg$ is negation, $\wedge$ is conjunction, $\Diamond$ is the *closure* operator, and $\mathcal{U}$ is the *until* operator. Closure (and interior, see Figure 2) operators come from the tradition of topological spatial logics [24].

**Definition 9.** *Satisfaction $\mathcal{M}, x \models \phi$ of formula $\phi$ at point $x$ in model $\mathcal{M} = ((X, \mathcal{C}), \mathcal{V})$ is defined, by induction on terms, as follows:*

$$\begin{array}{lll}
\mathcal{M}, x \models p & \iff & x \in \mathcal{V}(p) \\
\mathcal{M}, x \models \top & \iff & true \\
\mathcal{M}, x \models \neg\phi & \iff & \mathcal{M}, x \not\models \phi \\
\mathcal{M}, x \models \phi \wedge \psi & \iff & \mathcal{M}, x \models \phi \text{ and } \mathcal{M}, x \models \psi \\
\mathcal{M}, x \models \Diamond\phi & \iff & x \in \mathcal{C}(\{y \in X | \mathcal{M}, y \models \phi\}) \\
\mathcal{M}, x \models \phi \mathcal{U} \psi & \iff & \exists A \subseteq X . x \in A \wedge \forall y \in A . \mathcal{M}, y \models \phi \wedge \\
 & & \quad \wedge \forall z \in \mathcal{B}^+(A) . \mathcal{M}, z \models \psi
\end{array}$$

$$
\begin{array}{lll}
\bot \triangleq \neg\top & \phi \vee \psi \triangleq \neg(\neg\phi \wedge \neg\psi) & \Box\phi \triangleq \neg(\Diamond\neg\phi) \\
\partial\phi \triangleq (\Diamond\phi) \wedge (\neg\Box\phi) & \partial^{-}\phi \triangleq \phi \wedge (\neg\Box\phi) & \partial^{+}\phi \triangleq (\Diamond\phi) \wedge (\neg\phi) \\
\phi\,\mathcal{R}\psi \triangleq \neg((\neg\psi)\,\mathcal{U}(\neg\phi)) & \mathcal{G}\phi \triangleq \phi\,\mathcal{U}\bot & \mathcal{F}\phi \triangleq \neg\mathcal{G}(\neg\phi)
\end{array}
$$

**Fig. 2.** *SLCS* derivable operators

In Figure 2, we present some derived operators. Besides standard logical connectives, the logic can express the *interior* ($\Box\phi$), the *boundary* ($\partial\phi$), the *interior boundary* ($\partial^{-}\phi$) and the *closure boundary* ($\partial^{+}\phi$) of the set of points satisfying formula $\phi$. Moreover, by appropriately using the *until* operator, operators concerning *reachability* ($\phi\,\mathcal{R}\psi$), *global satisfaction* ($\mathcal{G}\phi$) and *possible satisfaction* ($\mathcal{F}\phi$) can be derived.

To clarify the expressive power of $\mathcal{U}$ and operators derived from it we provide Theorem 2 and Theorem 3, giving a formal meaning to the idea of "way out" of $\phi$, and providing an interpretation of $\mathcal{U}$ in terms of *paths*.

**Definition 10.** *A* closure-continuous function $f : (X_1, \mathcal{C}_1) \to (X_2, \mathcal{C}_2)$ *is a function* $f : X_1 \to X_2$ *such that, for all* $A \subseteq X_1$, $f(\mathcal{C}_1(A)) \subseteq \mathcal{C}_2(f(A))$.

**Definition 11.** *Consider a closure space* $(X, \mathcal{C})$*, and the quasi-discrete space* $(\mathbb{N}, \mathcal{C}_{Succ})$*, where* $(n, m) \in Succ \iff m = n{+}1$*. A (countable)* path *in* $(X, \mathcal{C})$ *is a closure-continuous function* $p : (\mathbb{N}, \mathcal{C}_{Succ}) \to (X, \mathcal{C})$*. We call* $p$ *a path* from $x$*, and write* $p : x \rightsquigarrow \infty$*, when* $p(0) = x$*. We write* $y \in p$ *whenever there is* $l \in \mathbb{N}$ *such that* $p(l) = y$*. We write* $p : x \overset{A}{\underset{y}{\rightsquigarrow}} \infty$ *when* $p$ *is a path from* $x$*, and there is* $l$ *with* $p(l) = y$ *and for all* $l' \leq l . p(l') \in A$.

**Theorem 2.** *If* $\mathcal{M}, x \models \phi\,\mathcal{U}\psi$*, then for each* $p : x \rightsquigarrow \infty$ *and* $l$*, if* $\mathcal{M}, p(l) \models \neg\phi$*, there is* $k \in \{1, \ldots, l\}$ *such that* $\mathcal{M}, p(k) \models \psi$.

Theorem 2 can be strengthened to a necessary and sufficient condition in the case of models based on quasi-discrete spaces. First, we establish that paths in a quasi-discrete space are also paths in its underlying graph.

**Lemma 3.** *Given path* $p$ *in a quasi-discrete space* $(X, \mathcal{C}_R)$*, for all* $i \in \mathbb{N}$ *with* $p(i) \neq p(i+1)$*, we have* $(p(i), p(i+1)) \in R$*, i.e., the image of* $p$ *is a (graph theoretical, infinite) path in the graph of* $R$*. Conversely, each path in the graph of* $R$ *uniquely determines a path in the sense of Definition 11.*

**Theorem 3.** *In a quasi-discrete closure model* $\mathcal{M}$*,* $\mathcal{M}, x \models \phi\,\mathcal{U}\psi$ *if and only if* $\mathcal{M}, x \models \phi$*, and for each path* $p : x \rightsquigarrow \infty$ *and* $l \in \mathbb{N}$*, if* $\mathcal{M}, p(l) \models \neg\phi$*, there is* $k \in \{1, \ldots, l\}$ *such that* $\mathcal{M}, p(k) \models \psi$.

```
Function Sat(M, φ)
    Input: Quasi-discrete closure model M = ((X, C), V), SLCS formula φ
    Output: Set of points {x ∈ X | M, x ⊨ φ}
    Match φ
        case ⊤ : return X
        case p : return V(p)
        case ¬ψ :
            let P = Sat(M, ψ) in
            return X \ P
        case ψ ∧ ξ :
            let P = Sat(M, ψ) in
            let Q = Sat(M, ξ) in
            return P ∩ Q
        case ◇ψ :
            let P = Sat(M, ψ) in
            return C(P)
        case ψ U ξ :  return CheckUntil (M,ψ, ξ)
```

**Algorithm 1:** Decision procedure for the model checking problem.

*Remark 3.* Directly from Theorem 3 and from the definitions in Figure 2 we have also that in a quasi-discrete closure model $\mathcal{M}$:

1. $\mathcal{M}, x \models \phi \mathcal{R} \psi$ iff. there is $p : x \rightsquigarrow \infty$ and $k \in \mathbb{N}$ such that $\mathcal{M}, p(k) \models \psi$ and for each $j \in \{1, \ldots, k\}$ $\mathcal{M}, p(j) \models \phi$;
2. $\mathcal{M}, x \models \mathcal{G}\phi$ iff. for each $p : x \rightsquigarrow \infty$ and $i \in \mathbb{N}$, $\mathcal{M}, p(i) \models \phi$;
3. $\mathcal{M}, x \models \mathcal{F}\phi$ iff. there is $p : x \rightsquigarrow \infty$ and $i \in \mathbb{N}$ such that $\mathcal{M}, p(i) \models \phi$.

Note that, a point $x$ satisfies $\phi \mathcal{R} \psi$ if and only if either $\psi$ is satisfied by $x$ or there exists a sequence of points after $x$, all satisfying $\phi$, leading to a point satisfying both $\psi$ and $\phi$. In the second case, it is not required that $x$ satisfies $\phi$.

## 5 Model checking *SLCS* formulas

In this section we present a model checking algorithm for $SLCS$, which is a variant of standard CTL model checking [3]. Function `Sat`, presented in Algorithm 1, takes as input a finite quasi-discrete model $\mathcal{M} = ((X, \mathcal{C}_R), \mathcal{V})$ and an $SLCS$ formula $\phi$, and returns the set of all points in $X$ satisfying $\phi$. The function is inductively defined on the structure of $\phi$ and, following a bottom-up approach, computes the resulting set via an appropriate combination of the recursive invocations of `Sat` on the sub-formulas of $\phi$. When $\phi$ is $\top$, $p$, $\neg\psi$ or $\psi \wedge \xi$, definition of $\texttt{Sat}(\mathcal{M}, \phi)$ is as expected.

```
Function CheckUntil (M,ψ, ξ)
    let V = Sat(M, ψ) in
    let Q = Sat(M, ξ) in
    var T := B⁺(V ∪ Q)
    while T ≠ ∅ do
        T' := ∅
        for x ∈ T do
            N := pre(x) ∩ V
            V := V \ N
            T' := T' ∪ (N \ Q)
        T := T';
    return V
```

**Algorithm 2:** Checking until formulas in a quasi-discrete closure space.

To compute the set of points satisfying $\Diamond\psi$, the closure operator $\mathcal{C}$ of the space is applied to the set of points satisfying $\psi$.

When $\phi$ is of the form $\psi\mathcal{U}\xi$, function `Sat` relies on the function `CheckUntil` defined in Algorithm 2. This function takes as parameters a model $\mathcal{M}$ and two $SLCS$ formulas $\psi$ and $\xi$ and computes the set of points in $\mathcal{M}$ satisfying $\psi\mathcal{U}\xi$ by removing from $V = \texttt{Sat}(\mathcal{M}, \psi)$ all the *bad* points. A point is *bad* if there exists a path passing through it, that leads to a point satisfying $\neg\psi$ without passing through a point satisfying $\xi$. Let $Q = \texttt{Sat}(\mathcal{M}, \xi)$ be the set of points in $\mathcal{M}$ satisfying $\xi$. To identify the *bad* points in $V$ the function `CheckUntil` performs a *backward search* from $T = \mathcal{B}^+(V \cup Q)$. Note that any *path* exiting from $V \cup Q$ has to pass through points in $T$. Moreover, the latter only contains points that satisfy neither $\psi$ nor $\xi$. Until $T$ is empty, function `CheckUntil` first picks an element $x$ in $T$ and then removes from $V$ the set of (bad) points $N$ that can reach $x$ in *one step*. To compute the set $N$ we use the function $pre(x) = \{y \in X \mid (y, x) \in R\}$.[6] At the end of each iteration the set $T$ is updated by considering the set of new discovered *bad points*.

**Lemma 4.** *Let $X$ a finite set and $R \subseteq X \times X$. For any finite quasi-discrete model $\mathcal{M} = ((X, \mathcal{C}_R), \mathcal{V})$ and $SLCS$ formula $\phi$ with $k$ operators,* `Sat` *terminates in $\mathcal{O}(k \cdot (|X| + |R|))$ steps.*

**Theorem 4.** *For any finite quasi-discrete closure model $\mathcal{M} = ((X, \mathcal{C}), \mathcal{V})$ and $SLCS$ formula $\phi$, $x \in \texttt{Sat}(\mathcal{M}, \phi)$ if and only if $\mathcal{M}, x \models \phi$.*

[6] Function *pre* can be *pre-computed* when the relation $R$ is loaded from the input.

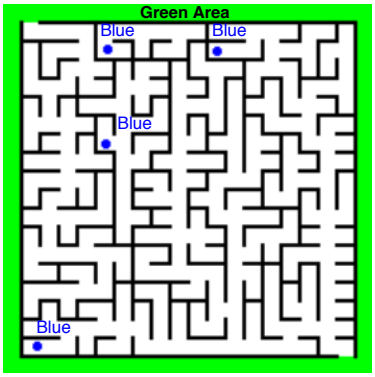


**Fig. 3.** A maze.

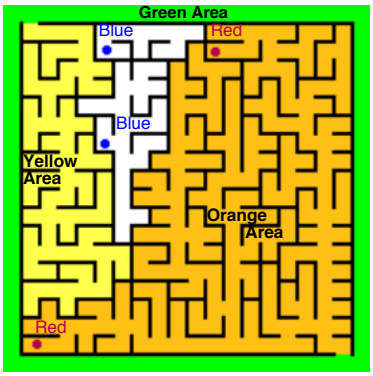


**Fig. 4.** Model checker output.

## 6 A model checker for spatial logics

The algorithm described in Section 5 is available as a proof-of-concept tool[7]. The tool, implemented using the functional language OCaml, contains a generic implementation of a global model-checker using closure spaces, parametrised by the type of models.

An example of the tool usage is to approximately identify regions of interest on a digital picture (e.g., a map, or a medical image), using spatial formulas. In this case, digital pictures are treated as quasi-discrete models in the plane $\mathbb{Z} \times \mathbb{Z}$. The language of propositions is extended to simple formulas dealing with colour ranges, in order to cope with images where there are different shades of certain colours.

In Figure 3 we show a digital picture of a maze. The green area is the exit. The blue areas are start points. The input of the tool is shown in Figure 5, where the `Paint` command is used to invoke the global model checker and colour points satisfying a given formula. Three formulas, making use of the until operator, are used to identify interesting areas. The output of the tool is in Figure 4. The colour red denotes start points from which the exit can be reached. Orange and yellow indicate the two regions through which the exit can be reached, including and excluding a start point, respectively.

In Figure 6 we show a digital image[8] depicting a portion of the map of Pisa, featuring a red circle which denotes a train station. Streets of different importance are painted with different colors in the map. The model checker is used to identify the area surrounding the station which

[7] Web site: `http://www.github.com/vincenzoml/slcs`.

[8] © *OpenStreetMap contributors* – `http://www.openstreetmap.org/copyright`.

```
Let reach(a,b) = !( (!b) U (!a) );
Let reachThrough(a,b) = a & reach((a|b),b);

Let toExit = reachThrough(["white"],["green"]);
Let fromStartToExit = toExit & reachThrough(["white"],["blue"]);
Let startCanExit = reachThrough(["blue"],fromStartToExit);

Paint "yellow" toExit;
Paint "orange" fromStartToExit;
Paint "red" startCanExit;
```

**Fig. 5.** Input to the model checker.


**Fig. 6.** Input: the map of a town.

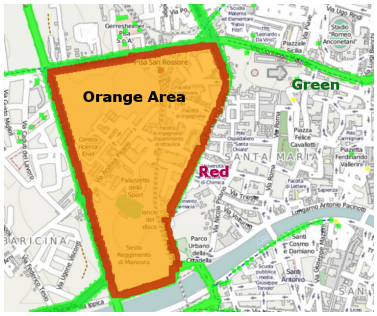


**Fig. 7.** Output of the tool.

is delimited by main streets, and the delimiting main streets. The output of the tool is shown in Figure 7, where the station area is coloured in orange, the surrounding main streets are red, and other main streets are in green. We omit the source code of the model checking session for space reasons (see the source code of the tool). As a mere hint on how practical it is to use a model checker for image analysis, the execution time on our test image, consisting of about 250000 pixels, is in the order of ten seconds on a standard laptop equipped with a 2Ghz processor.

## 7 Conclusions and Future Work

In this paper, we have presented a methodology to verify properties that depend upon space. We have defined an appropriate logic, stemming from the tradition of topological interpretations of modal logics, dating back to earlier logicians such as Tarski, where modalities describe neighbourhood. The topological definitions have been lifted to a more general setting, also encompassing discrete, graph-based structures. The proposed framework

has been extended with a spatial variant of the *until* operator, and we have also defined an efficient model checking procedure, which is implemented in a proof-of-concept tool.

As future work, we first of all plan to merge the results presented in this paper with temporal reasoning. This integration can be done in more than one way. It is not difficult to consider "snapshot" models consisting of a temporal model (e.g., a Kripke frame) where each state is in turn a closure model, and atomic formulas of the temporal fragment are replaced by spatial formulas. The various possible combinations of temporal and spatial operators, in linear and branching time, are examined for the case of topological models and basic modal formulas in [18]. Snapshot models may be susceptible to state-space explosion problems as spatial formulas could need to be recomputed at every state. On the other hand, one might be able to exploit the fact that changes of space over time are incremental and local in nature. Promising ideas are presented both in [17], where principles of "continuous change" are proposed in the setting of closure spaces, and in [20] where spatio-temporal models are generated by locally-scoped update functions, in order to describe dynamic systems. In the setting of collective adaptive systems, it will be certainly needed to extend the basic framework we presented with metric aspects (e.g., distance-bounded variants of the until operator), and probabilistic aspects, using atomic formulas that are probability distributions. A thorough investigation of these issues will be the object of future research.

A challenge in spatial and spatio-temporal reasoning is posed by recursive spatial formulas, *a la* $\mu$-calculus, especially on infinite structures with relatively straightforward generating functions (think of fractals, or fluid flow analysis of continuous structures). Such infinite structures could be described by topologically enhanced variants of $\omega$-automata. Classes of automata exist living in specific topological structures; an example is given by *nominal automata* (see e.g., [4,14,21]), that can be defined using presheaf toposes [13]. This standpoint could be enhanced with notions of neighbourhood coming from closure spaces, with the aim of developing a unifying theory of languages and automata describing space, graphs, and process calculi with resources.

## References


1. M. Aiello. *Spatial Reasoning: Theory and Practice*. PhD thesis, Institute of Logic, Language and Computation, University of Amsterdam, 2002.
2. M. Aiello, I. Pratt-Hartmann, and J. van Benthem, editors. *Handbook of Spatial Logics*. Springer, 2007.

3. C. Baier and J.P. Katoen. *Principles of model checking*. MIT Press, 2008.
4. M. Bojanczyk, B. Klin, and S. Lasota. Automata with group actions. In *LICS*, pages 355–364. IEEE Computer Society, 2011.
5. L. Bortolussi, J. Hillston, D. Latella, and M. Massink. Continuous approximation of collective system behaviour: A tutorial. *Perform. Eval.*, 70(5):317 – 349, 2013.
6. L. Caires and L. Cardelli. A spatial logic for concurrency (part I). *Information and Computation*, 186(2):194–235, 2003.
7. L. Cardelli, P. Gardner, and G. Ghelli. A spatial logic for querying graphs. In *ICALP*, volume 2380 of *LNCS*, pages 597–610. Springer, 2002.
8. L. Cardelli and A.D. Gordon. Anytime, anywhere: Modal logics for mobile ambients. In *POPL*, pages 365–377. ACM, 2000.
9. A. Chaintreau, J. Le Boudec, and N. Ristanovic. The age of gossip: Spatial mean field regime. SIGMETRICS, pages 109–120, New York, NY, USA, 2009. ACM.
10. Vincenzo Ciancia, Diego Latella, Michele Loreti, and Mieke Massink. Specifying and verifying properties of space - extended version. *CoRR*, abs/1406.6393, 2014.
11. R. De Nicola, G.L. Ferrari, and R. Pugliese. Klaim: A kernel language for agents interaction and mobility. *IEEE Trans. Software Eng.*, 24(5):315–330, 1998.
12. A. Del Bimbo, E. Vicario, and D. Zingoni. Symbolic description and visual querying of image sequences using spatio-temporal logic. *IEEE Trans. Knowl. Data Eng.*, 7(4):609–622, 1995.
13. M.P. Fiore and S. Staton. Comparing operational models of name-passing process calculi. *Information and Computation*, 204(4):524–560, 2006.
14. M.J. Gabbay and V. Ciancia. Freshness and name-restriction in sets of traces with names. In *FOSSACS*, volume 6604 of *LNCS*, pages 365–380. Springer, 2011.
15. F. Gadducci and A. Lluch-Lafuente. Graphical encoding of a spatial logic for the *pi* -calculus. In *CALCO*, volume 4624 of *LNCS*, pages 209–225. Springer, 2007.
16. A. Galton. The mereotopology of discrete space. In *COSIT*, volume 1661 of *LNCS*, pages 251–266. Springer, 1999.
17. A. Galton. A generalized topological view of motion in discrete space. *Theoretical Computer Science*, 305(1–3):111 – 134, 2003.
18. R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyaschev. Spatial logic + temporal logic = ? In Aiello et al. [2], pages 497–564.
19. V.A. Kovalevsky. *Geometry of Locally Finite Spaces: Computer Agreeable Topology and Algorithms for Computer Imagery*. House Dr. Baerbel Kovalevski, 2008.
20. P. Kremer and G. Mints. Dynamic topological logic. In Aiello et al. [2], pages 565–606.
21. A. Kurz, T. Suzuki, and E. Tuosto. On nominal regular languages with binders. In *FoSSaCS*, volume 7213 of *LNCS*, pages 255–269. Springer, 2012.
22. A. Rosenfeld. Digital topology. *The American Mathematical Monthly*, 86(8):621–630, 1979.
23. M.B. Smyth and J. Webster. Discrete spatial models. In Aiello et al. [2], pages 713–798.
24. J. van Benthem and G. Bezhanishvili. Modal logics of space. In *Handbook of Spatial Logics*, pages 217–298. 2007.
25. T. Yung Kong and A. Rosenfeld. Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48(3):357–393, 1989.