Embedding RCC8D in the Collective Spatial Logic CSLCS

(corrected version)

Vincenzo Ciancia[1], Diego Latella[1], and Mieke Massink[1]

Consiglio Nazionale delle Ricerche - Istituto di Scienza e Tecnologie dell'Informazione
'A. Faedo', CNR, Italy

Abstract. Discrete mereotopology is a logical theory for the specification of qualitative spatial functions and relations defined over a discrete space, intended as a set of basic elements, the *pixels*, with an adjacency relation defined over it. The notions of interest are that of *region*, intended as an arbitrary aggregate of pixels, and of specific *relations* between regions. The mereotopological theory RCC8D extends the mereological theory RCC5D—a theory of region parthood for discrete spaces—with the topological notion of *connection* and the remaining relations (disconnection, external connection, tangential and nontangential proper parthood and their inverses). In this paper, we propose an encoding of RCC8D into CSLCS, the *collective* extension of the *Spatial Logic of Closure Spaces* SLCS. We show how `topochecker`, a model-checker for CSLCS, can be used for effectively checking the existence of a RCC8D relation between two given regions of a discrete space.

Keywords: RCC8D, Adjacency Spaces, Closure Spaces, Spatial Logics, SLCS, CSLCS.

1 Introduction

The study of logical approaches to modelling *space* and spatial aspects of computation is a well established area of research in computer science and artificial intelligence. A standard reference is the *Handbook of Spatial Logics* [1]. Therein, several spatial logics are described, with applications far beyond topological spaces; such logics treat not only aspects of morphology, geometry and distance, but also advanced topics such as dynamic systems, and discrete structures, that are particularly difficult to deal with, especially from a topological perspective (see, for example [15,19]). For this reason, most of the work present in the literature deals with continuous notions of space, such as Euclidean spaces. In this context, a prominent area of research is represented by the logical theories of "parthood"—*Mereology*—and of "connection" between "regions", i.e. sets of points in a continuous space—*Mereotopology*—representative of which are the Region Connection Calculi RCC5 and RCC8, respectively. In particular, RCC8 [17] is widely referred to in the AI literature on Qualitative Spatial Reasoning [5].

More recently, attention has been devoted also to logical approaches to *discrete* spaces, including e.g. graphs or digital images, given the importance of such structures in computer science. In particular, in [18] the notions of *Discrete Mereology* and *Discrete Meretopology* have been presented and discrete versions of RCC5 and RCC8, namely RCC5D and RCC8D, have been defined.

On the other hand, in recent work [9,10], Ciancia et al. proposed the *Spatial Logic for Closure Spaces* (SLCS), defined along the same lines as the classical work of Tarski on the spatial interpretation of the modal *possibility* operator as the topological *closure* operator, but with two major differences. The first one is that the underlying model for the logic is not that of topological spaces, as in the classical approach, but rather *Closure Spaces* [15,16], a generalisation of topological spaces including also discrete structures such as graphs, and, consequently, digital images. The second one is the inclusion of the *surrounded* operator—denoted by $\mathcal{S}$, to be read "surrounded"—an operator similar to the *spatial until* discussed in [20] in the context of continuous spaces; a point satisfies $\Phi_1 \ \mathcal{S} \ \Phi_2$ if it satisfies Φ_1 and there is no way for moving away to a point not satisfying Φ_1 without first passing by a point satisfying Φ_2. In other words, the points satisfying Φ_1 are *surrounded by* points satisfying Φ_2. In addition, in [10] the logic has been extended with the *collective* fragment, leading to the definition of the *Collective Spatial Logic for Closure Spaces* (CSLCS), where properties of (connected) *sets* of points can be specified. Efficient model checking algorithms have been defined for both SLCS and CSLCS and have been implemented in the prototype tool `topochecker`[1].

In this paper we present an encoding of RCC8D into CSLCS. This shows that CSLCS is a suitable logic not only for reasoning about points in (closure) spaces and connected sets of such points, but also for regions in the sense of the Region Calculus and, in particular, of RCC8D.

The paper is organised as follows: in Section 2, SLCS and its extension CSLCS are briefly described; furthermore, we state a proposition relating the temporal *weak until* connective with the interpretation of spatial *surrounded* on discrete spatial models—the proof is provided in the appendix. Section 3 recalls Adjacency Spaces and RCC8D of [5]. The encoding procedure is described in Section 4 where some examples of use of `topochecker` are also shown as well as the (graphical) result of RCC8D relations over sample regions. Finally, in Section 6 some conclusions are drawn.

2 Spatial Logics for Closure Spaces

Spatial logics have been mainly studied from the point of view of *modal* logics. In his seminal work of 1938, Tarski presented a spatial, and in particular topological, interpretation of modal logic; in 1944 Tarski and McKinsey proved that the simple (and decidable) modal system $\mathcal{S}4$ is complete when interpreting the *possibility* modality $\Diamond$ of $\mathcal{S}4$ as *closure* on the reals or any similar metric

[1] Topochecker: *a topological model checker*, see `http://topochecker.isti.cnr.it`, `https://github.com/vincenzoml/topochecker`

space. More specifically, a topological model $\mathcal{M} = ((X, O), \mathcal{V})$ of modal logic is any topological space (X, O) where each *point* $x \in X$ is associated with the set of *atomic propositions* p it satisfies, namely the set $\{p | x \in \mathcal{V}(p)\}$, negation and conjunction are interpreted in the usual way, and the *possibility* operator $\Diamond$ is interpreted as *topological closure*, as follows (see [1], Chapter 5):

$$\begin{aligned} \mathcal{M}, x \models \Diamond \Phi \Leftrightarrow & \textit{ for all open sets } o \in O \textit{ such that } x \in o \\ & \textit{ there exists } x' \in o \textit{ such that } \mathcal{M}, x' \models \Phi. \end{aligned}$$

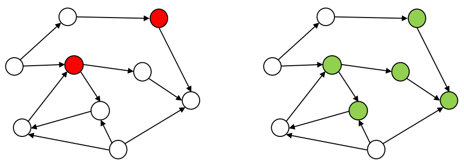
Of course, by duality, the *necessity* operator $\Box$ turns out to be interpreted as the *topological interior* operator, namely $\mathcal{M}, x \models \Box\Phi \Leftrightarrow$ *there exists a open set* $o \in O$ *such that* $x \in o$ *and* $\mathcal{M}, x' \models \Phi$ *for all* $x' \in o$. We refer the reader to [20] for further details. A legitimate question is whether the restriction to topological spaces is too strong. For answering this question, it is appropriate to focus on *discrete* spaces, e.g. graphs; any logical approach to reasoning about spatial properties of distributed systems should obviously be capable to deal with discrete structures. There exist of course relational models of $\mathcal{S}4$, namely reflexive and transitive Kripke structures and it is possible to derive a topological space from any such a structure in a sound and complete way. The topological spaces that are used are the so-called *Alexandroff spaces*. These are topological spaces in which each point has a least open neighbourhood. Unfortunately, the correspondence between topological spaces and reflexive and transitive Kripke structures is not easily extended to arbitrary Kripke structures, as transitivity and reflexivity always hold in topo-logics where the basic modality is the closure. On the other hand, requiring transitivity in all models may be too limiting a constraint. This is the main reason to further investigate non-transitive concepts of spatial models and for resorting to models which are more general than topological spaces. In our approach we use *closure spaces* as a generalisation of topological spaces.

Definition 1. *A* closure space *is a pair* $(X, \mathcal{C})$ *where* X *is a non-empty set (of* points*) and* $\mathcal{C} : 2^X \to 2^X$ *is a function satisfying the following axioms:*

1. $\mathcal{C}(\emptyset) = \emptyset$;
2. $Y \subseteq \mathcal{C}(Y)$ *for all* $Y \subseteq X$;
3. $\mathcal{C}(Y_1 \cup Y_2) = \mathcal{C}(Y_1) \cup \mathcal{C}(Y_2)$ *for all* $Y_1, Y_2 \subseteq X$. •

It is worth pointing out that topological spaces coincide with the sub-class of closure spaces for which also the *idempotence* axiom $\mathcal{C}(\mathcal{C}(Y)) = \mathcal{C}(Y)$ holds.

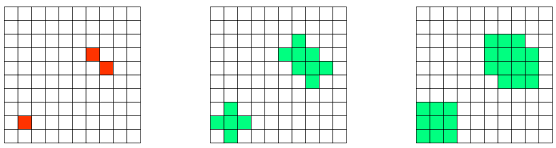
Given any relation $R \subseteq X \times X$, function $\mathcal{C}_R : 2^X \to 2^X$ with $\mathcal{C}_R(Y) \triangleq Y \cup \{x | \exists y \in Y. y\, R\, x\}$ satisfies the axioms of Definition 1 thus making $(X, \mathcal{C}_R)$ a closure space. It can be shown that the sub-class of closure spaces that can be generated by a relation as above coincides with the class of *quasi-discrete* closure spaces, i.e. closure spaces where every $x \in X$ has a minimal neighbourhood or, equivalently, for each $Y \subseteq X, \mathcal{C}(Y) = \bigcup_{y \in Y} \mathcal{C}(\{y\})$. Thus (finite) discrete structures, like graphs or Kripke structures can be (re-)interpreted as *quasi-discrete* closure spaces. For example, consider the graph of Figure 1 where a set Y of nodes is shown in red (1a); the closure $\mathcal{C}(Y)$ of Y is shown in green (1b).



(a) (b)

Fig. 1: A set of nodes Y in a graph (1a) and its closure $\mathcal{C}(Y)$ (1b).

Being a special case of graphs, also digital images can be modelled by (finite) quasi-discrete closure spaces. In particular, the pixels of the image are the nodes of the space, whereas the relevant relation is typically both reflexive and symmetric. It may relate any pixel with all the pixels with which it shares an edge, i.e. 5 pixels in 2D images, or with all the pixels with which it shares an edge or a corner, i.e. 9 pixels in 2D images. In the first case, the relation is called *othogonal*, whereas in the second case it is called *orthodiagonal*; in Section 3 we will use the orthodiagonal relation, also called the *adjacency relation* in [18]. For instance, the closure of the set of red pixels Y in Figure 2a is shown in green in Figure 2b, where the orthogonal relation is used, and in Figure 2c, where the orthodiagonal relation is used instead.



(a) (b) (c)

Fig. 2: A set of red pixels Y in a digital image (2a) and its closure $\mathcal{C}(Y)$ according to the orthogonal relation (2b) and the orthodiagonal relation (2c).

The hierarchy of closure spaces is shown in Figure 3.

2.1 The Spatial Logic for Closure Spaces - SLCS

In [9,10] the Spatial Logic for Closure Spaces (SLCS) was proporsed. In the remainder of this section we briefly recall the fragment of the logic we use in

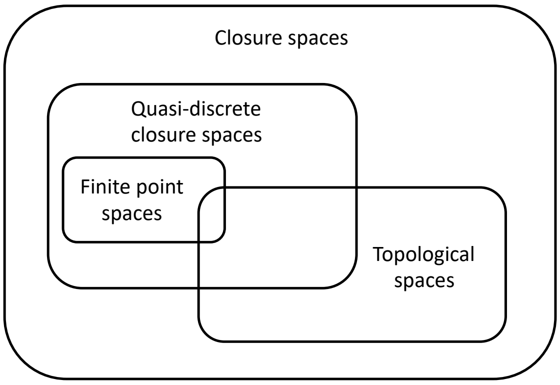


Fig. 3: The hierarchy of closure spaces.

the present paper, which consists essentially of $\mathcal{S}4$—where the $\Diamond$ operator is renamed $\mathcal{N}$ (to be read as *near*) for clarity reasons—enriched with an additional operator, the *surrounded* operator $\mathcal{S}$, where $\Phi_1 \, \mathcal{S} \, \Phi_2$ characterises the set of points belonging to an area satisfying Φ_1 and such that one cannot "escape" from such an area without hitting a point satisfying Φ_2, i.e. they are *surrounded* by Φ_2. The syntax of SLCS is given below, for P a set of *atomic predicates* p:

$$\Phi ::= p \mid \neg \Phi \mid \Phi_1 \vee \Phi_2 \mid \mathcal{N}\Phi \mid \Phi_1 \, \mathcal{S} \, \Phi_2 \tag{1}$$

In the sequel we provide a formal definition of the satisfaction relation for SLCS. To that purpose, we need to first introduce the notion of path. A (quasi-discrete) *path* π in $(X, \mathcal{C}_R)$ is a function $\pi : \mathbb{N} \to X$, such that for all $Y \subseteq \mathbb{N}$, $\pi(\mathcal{C}_{Succ}(Y)) \subseteq \mathcal{C}_R(\pi(Y))$, where $\pi(Y)$ is the pointwise extension of π on a set of points Y and $(\mathbb{N}, \mathcal{C}_{Succ})$ is the closure space of the natural numbers with the *successor* relation: $(n, m) \in Succ \Leftrightarrow m = n + 1$. Informally: the ordering in the path imposed by $\mathbb{N}$ is compatible with relation R, i.e. $\pi(i) \, R \, \pi(i + 1)$. Technically, a (quasi-discrete) path is a *continuous* function from $(\mathbb{N}, \mathcal{C}_{Succ})$ to $(X, \mathcal{C}_R)$. We refer to [10] for details. Set $Y \subseteq X$ is *path-connected* if for all points $y_1, y_2 \in Y$ there exists a path π and an index i such that: $\pi(0) = y_1$, $\pi(i) = y_2$ and $\pi(j) \in Y$, for all $0 \leq j \leq i$.

Definition 2. *A* closure model $\mathcal{M}$ *is a tuple* $\mathcal{M} = ((X, \mathcal{C}), \mathcal{V})$*, where* $(X, \mathcal{C})$ *is a closure space and* $\mathcal{V} : P \to 2^X$ *is a valuation assigning to each atomic predicate the set of points where it holds.* •

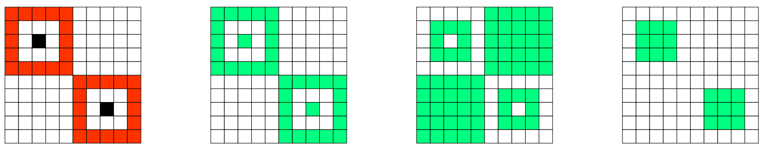
Definition 3. Satisfaction $\mathcal{M}, x \models \Phi$ *of a formula* Φ *at point* $x \in X$ *in model* $\mathcal{M} = ((X, \mathcal{C}), \mathcal{V})$ *is defined by induction on the structure of formulas:*

$$
\begin{array}{ll}
\mathcal{M}, x \models p \in P & \Leftrightarrow x \in \mathcal{V}(p) \\
\mathcal{M}, x \models \neg \Phi & \Leftrightarrow \mathcal{M}, x \models \Phi \textit{ does not hold} \\
\mathcal{M}, x \models \Phi_1 \vee \Phi_2 & \Leftrightarrow \mathcal{M}, x \models \Phi_1 \textit{ or } \mathcal{M}, x \models \Phi_2 \\
\mathcal{M}, x \models \mathcal{N} \Phi & \Leftrightarrow x \in \mathcal{C}(\{y | \mathcal{M}, y \models \Phi\}) \\
\mathcal{M}, x \models \Phi_1 \, \mathcal{S} \, \Phi_2 & \Leftrightarrow \mathcal{M}, x \models \Phi_1 \textit{ and} \\
& \quad \textit{for all paths } \pi \textit{ and indexes } \ell \textit{ the following holds:} \\
& \quad\quad \pi(0) = x \textit{ and } \mathcal{M}, \pi(\ell) \models \neg \Phi_1 \\
& \quad\quad \textit{implies} \\
& \quad\quad \textit{there exists index } j \textit{ such that:} \\
& \quad\quad\quad 0 < j \leq \ell \textit{ and } \mathcal{M}, \pi(j) \models \Phi_2
\end{array}
$$

•

Standard derived operators can be defined in the usual way e.g.: $\Phi_1 \wedge \Phi_2 \equiv \neg(\neg\Phi_1 \vee \neg\Phi_2)$, $\top \equiv p \vee \neg p$, $\bot \equiv \neg\top$, and so on.

In Figure 4a an example is shown of a model, based on a 2D space of 100 points arranged as a 10×10 grid, with reflexive, symmetric and orthogonal relation. We assume the set of atomic predicates P is the set $\{\mathit{black}, \mathit{white}, \mathit{red}\}$ and, in Figure 4a, we color in black the points satisfing *black* and similary for *white* and *red*. In Figure 4b the points satisfying formula $\mathit{black} \vee \mathit{red}$ are shown in green[2]; similarly, Figure 4c shows the points satisfying $\neg(\mathit{black} \vee \mathit{red})$, and Figure 4d shows those satisfying $\mathcal{N}\,\mathit{black}$. Finally, the points in Figure 4a satisfying *black* satisfy also $\mathit{black}\;\mathcal{S}(\mathcal{N}\,\mathit{red})$. Several examples of use of SLCS, extensions thereof, and related model-checking tools can be found in [9,10,13,3,12,14,11]



(a) (b) (c) (d)

Fig. 4: An example model (4a); the points shown in green are those satisfying $\mathit{black} \vee \mathit{red}$ (4b), $\neg(\mathit{black} \vee \mathit{red})$ (4c), and $\mathcal{N}\,\mathit{black}$ (4d).

Finally, we show the formal relationship between the SLCS *surrounded* operator interpreted on quasi-discrete closure spaces and the temporal logic *weak until* operator. Let us consider a set X and a relation $R \subseteq X \times X$; the pair $(X, \mathcal{C}_R)$ is a quasi-discrete closure space, but also a *Kripke frame*; any valuation $\mathcal{V}$ of atomic propositions makes such a space (frame) a closure model (Kripke model). The until operator $\Phi_1 \mathcal{U} \Phi_2$ is well-known. Let us recall the *weak until* operator $\Phi_1 \, \mathcal{W} \, \Phi_2$, whose satisfaction for path π is defined as $\mathcal{M}, \pi \models \Phi_1 \, \mathcal{W} \, \Phi_2$ iff $\mathcal{M}, \pi(i) \models \Phi_1$ for all i, or $\mathcal{M}, \pi \models \Phi_1 \, \mathcal{U} \, \Phi_2$ (note that $\mathcal{W}$ and $\mathcal{U}$ are path-formulas). The following holds:

[2] Note that this colour does *not* correspond to any atomic predicate and so it is not part of the model; we use it only for illustration purposes.

Proposition 1.

$$\Phi_2 \vee (\Phi_1 \,\mathcal{S}\, \Phi_2) \equiv A(\Phi_1 \,\mathcal{W}\, \Phi_2)$$

where A is the *path universal quantifier*. The proof is provided in the Appendix.

2.2 The Collective Extension - CSLCS

In this section we show how the logic defined above is extended in order to reason about *sets* of (connected) points, instead of individual points (see [10] for details). We introduce an additional class of formulas, namely the *collective formulas* by extending the grammar given in (1) as follows:

$$\Psi ::= \neg\Psi \mid \Psi_1 \wedge \Psi_2 \mid \Phi \prec \Psi \mid \mathcal{G}\Phi \tag{2}$$

Let Φ be an SLCS formula ("individual" formula, in the sequel), and Ψ a collective formula. Informally, $\Phi \prec \Psi$ (read: Φ *share* Ψ) is satisfied by set Y when the subset of points of Y satisfying the individual property Φ also satisfies the collective property Ψ. Formula $\mathcal{G}\Phi$ (read: *group* Φ) holds on set Y when the elements of the latter belong to a *group*, that is, a possibly larger, path-connected set of points, all satisfying the individual formula Φ. The satisfaction relation $\models_C$ for CSLCS is defined below:

Definition 4. Satisfaction $\mathcal{M}, Y \models_C \Psi$ *of a collective formula* Ψ *at set* $Y \subseteq X$ *in model* $\mathcal{M} = ((X, \mathcal{C}), \mathcal{V})$ *is defined by induction on the structure of formulas:*

$$\begin{array}{lll}
\mathcal{M}, Y \models_C \neg\Psi & \Leftrightarrow & \mathcal{M}, Y \models_C \Psi \textit{ does not hold} \\
\mathcal{M}, Y \models_C \Psi_1 \wedge \Psi_2 & \Leftrightarrow & \mathcal{M}, Y \models_C \Psi_1 \textit{ and } \mathcal{M}, Y \models_C \Psi_2 \\
\mathcal{M}, Y \models_C \Phi \prec \Psi & \Leftrightarrow & \mathcal{M}, \{x \in Y | \mathcal{M}, x \models \Phi\} \models_C \Psi \\
\mathcal{M}, Y \models_C \mathcal{G}\Phi & \Leftrightarrow & \textit{there exists } Z \subseteq X \textit{ such that} \\
 & & Y \subseteq Z \textit{ and } Z \textit{ is path-connected and} \\
 & & \textit{for all } z \in Z \textit{ we have: } \mathcal{M}, z \models \Phi
\end{array}$$

•

Back to Figure 4a, we note that, although *each* point satisfying *black* satisfies also $(black \vee white)\mathcal{S}red$, *the set* consisting exactly of the two points satisfying *black* does *not* satisfy the collective formula $\mathcal{G}((black \vee white)\mathcal{S}red)$, i.e. the members of the set are not surrounded *collectively* by red points. The set of black points in Figure 5 instead satisfies $\mathcal{G}((black \vee white)\mathcal{S}red)$.

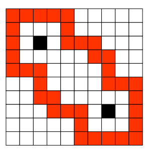


Fig. 5: A model where the set of the black points satisfies $\mathcal{G}((black \vee white)\mathcal{S}red)$.

Finally, it is useful to note that $\mathcal{M}, Y \models_C \Phi \prec \mathcal{G}\bot$ for every $\mathcal{M}$ and every Y if and only if $\Phi \equiv \bot$. Thus the formula $\Phi \prec \mathcal{G}\bot$ can be used for checking wether Φ denotes the empty set.

3 Discrete Spaces with Adjacency and RCC8D

In this section we briefly introduce a subclass of quasi-discrete closure spaces, namely those spaces $(X, \mathcal{C}_R)$ where the underlying relation R, called the *adjacency relation*, is *reflexive* and *symmetric*. The points of any such space can be thought of as *pixels* and the space itself can be used as (a model for) a digital picture [18].

Discrete Mereotopology (DM) is concerned with the study of the relations among *regions*, where a region is interpreted as an arbitrary aggregate $Y \subseteq X$ of pixels. In particular, Mereology is the theory of *parthood* and those relations which can be defined in terms of it. Parthood is defined as set inclusion restricted to non-null regions:

$$\mathtt{P}(Y_1, Y_2) \equiv_{\text{def}} Y_1 \subseteq Y_2 \text{ and } Y_1 \neq \emptyset.$$

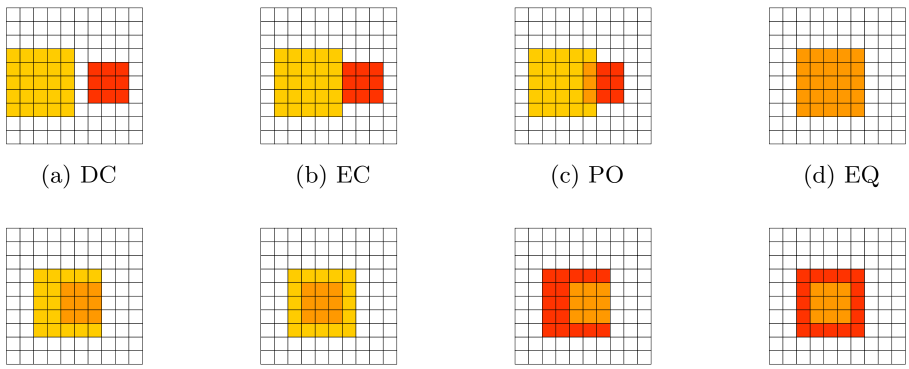
The intuition behind the definition of $\mathtt{P}(Y_1, Y_2)$ is fairly simple and comes from set theory: Y_1 is part of Y_2 and should not be empty. The derived relations are defined below. They are readily explained in terms of set theory; the interested reader is referred to [18] for a discussion on the region relations and on their relationships:

$\mathtt{PP}(Y_1, Y_2)$	$\equiv_{\text{def}} \mathtt{P}(Y_1, Y_2) \wedge Y_1 \neq Y_2$	[Proper Parthood]
$\mathtt{Pi}(Y_1, Y_2)$	$\equiv_{\text{def}} \mathtt{P}(Y_2, Y_1)$	[Inverse Parthood]
$\mathtt{PPi}(Y_1, Y_2)$	$\equiv_{\text{def}} \mathtt{PP}(Y_2, Y_1)$	[Inverse Proper Parthood]
$\mathtt{O}(Y_1, Y_2)$	$\equiv_{\text{def}} Y_2 \cap Y_1 \neq \emptyset$	[Overlap]
$\mathtt{PO}(Y_1, Y_2)$	$\equiv_{\text{def}} \mathtt{O}(Y_1, Y_2) \wedge \neg\mathtt{P}(Y_1, Y_2) \wedge \neg\mathtt{P}(Y_2, Y_1)$	[Partially Overlap]
$\mathtt{DR}(Y_1, Y_2)$	$\equiv_{\text{def}} \neg\mathtt{O}(Y_1, Y_2)$	[Discrete]
$\mathtt{EQ}(Y_1, Y_2)$	$\equiv_{\text{def}} \mathtt{P}(Y_1, Y_2) \wedge \mathtt{P}(Y_2, Y_1)$	[Equal]

The relation set $\{\mathtt{DR}, \mathtt{PO}, \mathtt{PP}, \mathtt{PPi}, \mathtt{EQ}\}$ is referred to as RCC5D, i.e. the Discrete Region Connection Calculus based on 5 relations, which is a purely mereological language. It is extended to the mereotopological language RCC8D through the addition of the topological notion of *connection* and operators derived thereof, as follows:

$\mathtt{C}(Y_1, Y_2)$	$\equiv_{\mathrm{def}} \exists y_1 y_2 (y_1 \in Y_1 \wedge y_2 \in Y_2 \wedge y_1 R y_2)$	[CONNECTION]
$\mathtt{DC}(Y_1, Y_2)$	$\equiv_{\mathrm{def}} \neg \mathtt{C}(Y_1, Y_2)$	[DISCONNECTION]
$\mathtt{EC}(Y_1, Y_2)$	$\equiv_{\mathrm{def}} \mathtt{C}(Y_1, Y_2) \wedge \neg \mathtt{O}(Y_1, Y_2)$	[EXTERNAL CONNECTION]
$\mathtt{TPP}(Y_1, Y_2)$	$\equiv_{\mathrm{def}} \mathtt{PP}(Y_1, Y_2) \wedge \exists Z (\mathtt{EC}(Z, Y_1) \wedge \mathtt{EC}(Z, Y_2))$	[TANGENTIAL PARTHOOD]
$\mathtt{NTPP}(Y_1, Y_2)$	$\equiv_{\mathrm{def}} \mathtt{PP}(Y_1, Y_2) \wedge \neg \exists Z (\mathtt{EC}(Z, Y_1) \wedge \mathtt{EC}(Z, Y_2))$	[NON TANGENTIAL PARTHOOD]
$\mathtt{TPPi}(Y_1, Y_2)$	$\equiv_{\mathrm{def}} \mathtt{TPP}(Y_2, Y_1)$	[INV. TANGENTIAL PARTHOOD]
$\mathtt{NTPPi}(Y_1, Y_2)$	$\equiv_{\mathrm{def}} \mathtt{NTPP}(Y_2, Y_1)$	[INV. NON TANG. PARTHOOD]

The relation set $\{\mathtt{DC}, \mathtt{EC}, \mathtt{PO}, \mathtt{TPP}, \mathtt{NTPP}, \mathtt{TPPi}, \mathtt{NTPPi}, \mathtt{EQ}\}$ forms what is known as RCC8D. In Figure 6 we give an illustration of these relations using models based on a 2D space of 100 points arranged as a 10×10 grid, with reflexive, symmetric and orthodiagonal relation, as in [18], which we refer to for a more detailed description.



(a) DC (b) EC (c) PO (d) EQ

(e) TPP (f) NTPP (g) TPPi (h) NTPPi

Fig. 6: The eight RCC8D relations.

4 Encoding RCC8D into CSLCS

Let us now focus on the encoding of RCC8D in CSLCS. Let $(X, \mathcal{C})$ be a finite closure space. We associate the atomic predicate p_Y to each set $Y \subseteq X$, such that in all closure models $\mathcal{M} = ((X, \mathcal{C}), \mathcal{V})$ we have $\mathcal{V}(p_Y) = Y$. The encoding $[\![\cdot]\!]$ of RCC8D in CSLCS is defined in the sequel.

We first encode standard set theoretic and closure operations into CSLCS in the obvious way; in the sequel $\gamma, \gamma_1, \gamma_2$ range over expressions on sets built out of constants, complement, intersection and closure:

$$
\begin{array}{lll}
\llbracket Y \rrbracket & = p_Y, \text{for all } Y \subseteq X & [\textsc{Constant}] \\
\llbracket \overline{\gamma} \rrbracket & = \neg \llbracket \gamma \rrbracket & [\textsc{Complement}] \\
\llbracket \gamma_1 \cap \gamma_2 \rrbracket & = \llbracket \gamma_1 \rrbracket \wedge \llbracket \gamma_2 \rrbracket & [\textsc{Intersection}] \\
\llbracket \mathcal{C}(\gamma) \rrbracket & = \mathcal{N}(\llbracket \gamma \rrbracket) & [\textsc{Closure}]
\end{array}
$$

Now we add the tests on the empty set, on set-inclusion and set-equality; note the use of the format $\Phi \prec \mathcal{G}\bot$ to check for the empty set, discussed at the end of Section 2:

$$
\begin{array}{lll}
\llbracket \gamma = \emptyset \rrbracket & = \llbracket \gamma \rrbracket \prec \mathcal{G}\bot & [\textsc{Empty}] \\
\llbracket \gamma_1 \subseteq \gamma_2 \rrbracket & = \llbracket (\gamma_1 \cap \overline{\gamma_2}) = \emptyset \rrbracket & [\textsc{Inclusion}] \\
\llbracket \gamma_1 = \gamma_2 \rrbracket & = \llbracket \gamma_1 \subseteq \gamma_2 \rrbracket \wedge \llbracket \gamma_2 \subseteq \gamma_1 \rrbracket & [\textsc{Equality}]
\end{array}
$$

Finally, the actual encoding of (RCC5D and) RCC8D is given below and is self-explanatory; the right-hand side of the equation for the encoding of a relation is just the logical encoding of the set-theoretical expression used in the definition of the relation presented in [18] and recalled in Section 3 of the present paper:

$$
\begin{array}{lll}
\llbracket \mathtt{P}(Y_1, Y_2) \rrbracket & = \llbracket Y_1 \subseteq Y_2 \rrbracket \wedge \neg \llbracket Y_1 = \emptyset \rrbracket & [\textsc{Parthood}] \\
\llbracket \mathtt{PP}(Y_1, Y_2) \rrbracket & = \llbracket \mathtt{P}(Y_1, Y_2) \rrbracket \wedge \neg \llbracket Y_1 = Y_2 \rrbracket & [\textsc{Proper Parthood}] \\
\llbracket \mathtt{Pi}(Y_1, Y_2) \rrbracket & = \llbracket \mathtt{P}(Y_2, Y_1) \rrbracket & [\textsc{Inverse Parthood}] \\
\llbracket \mathtt{PPi}(Y_1, Y_2) \rrbracket & = \llbracket \mathtt{PP}(Y_2, Y_1) \rrbracket & [\textsc{Inverse Proper Parthood}] \\
\llbracket \mathtt{O}(Y_1, Y_2) \rrbracket & = \neg \llbracket Y_1 \cap Y_2 = \emptyset \rrbracket & [\textsc{Overlap}] \\
\llbracket \mathtt{PO}(Y_1, Y_2) \rrbracket & = \llbracket \mathtt{O}(Y_1, Y_2) \rrbracket \wedge \neg \llbracket \mathtt{P}(Y_1, Y_2) \rrbracket \wedge \neg \llbracket \mathtt{P}(Y_2, Y_1) \rrbracket & [\textsc{Partial Overlap}] \\
\llbracket \mathtt{DR}(Y_1, Y_2) \rrbracket & = \neg \llbracket \mathtt{O}(Y_1, Y_2) \rrbracket & [\textsc{Discrete}] \\
\llbracket \mathtt{EQ}(Y_1, Y_2) \rrbracket & = \llbracket \mathtt{P}(Y_1, Y_2) \rrbracket \wedge \llbracket \mathtt{P}(Y_2, Y_1) \rrbracket & [\textsc{Equality on non-null Regions}] \\
\llbracket \mathtt{C}(Y_1, Y_2) \rrbracket & = \neg (\llbracket \mathcal{C}(Y_1) \cap Y_2 = \emptyset \rrbracket \vee \llbracket \mathcal{C}(Y_2) \cap Y_1 = \emptyset \rrbracket) & [\textsc{Connection}] \\
\llbracket \mathtt{DC}(Y_1, Y_2) \rrbracket & = \neg \llbracket \mathtt{C}(Y_1, Y_2) \rrbracket & [\textsc{Disconnection}] \\
\llbracket \mathtt{EC}(Y_1, Y_2) \rrbracket & = \llbracket \mathtt{C}(Y_1, Y_2) \rrbracket \wedge \neg \llbracket \mathtt{O}(Y_1, Y_2) \rrbracket & [\textsc{External connection}] \\
\llbracket \mathtt{TPP}(Y_1, Y_2) \rrbracket & = \llbracket \mathtt{PP}(Y_1, Y_2) \rrbracket \wedge \neg \llbracket \mathcal{C}(Y_1) \cap \overline{Y_2} = \emptyset \rrbracket & [\textsc{Tangential PP}] \\
\llbracket \mathtt{NTPP}(Y_1, Y_2) \rrbracket & = \llbracket \mathtt{PP}(Y_1, Y_2) \rrbracket \wedge \llbracket \mathcal{C}(Y_1) \cap \overline{Y_2} = \emptyset \rrbracket & [\textsc{Nontangential PP}] \\
\llbracket \mathtt{TPPi}(Y_1, Y_2) \rrbracket & = \llbracket \mathtt{TPP}(Y_2, Y_1) \rrbracket & [\textsc{Inverse Tangential PP}] \\
\llbracket \mathtt{NTPPi}(Y_1, Y_2) \rrbracket & = \llbracket \mathtt{NTPP}(Y_2, Y_1) \rrbracket & [\textsc{Inverse Nontangential PP}]
\end{array}
$$

Correctness of the above encoding is stated below:

Proposition 2. *For all RCC8D formulas F the following holds: F holds in an adjacency model $\mathcal{M} = ((X, \mathcal{C}), \mathcal{V})$ if and only if $\mathcal{M}, X \models_C \llbracket F \rrbracket$.*

Proof. The proposition is straightforward to prove. The only case which requires a bit of explanation concerns the TPP predicate (and NTPP). The definition of TPP given in [18] is the following:

$$\mathtt{TPP}(Y_1, Y_2) = \mathtt{PP}(Y_1, Y_2) \land \exists Z.(\mathtt{EC}(Z, Y_1) \land \mathtt{EC}(Z, Y_2)).$$

We show that the two definitions characterise the same property. Note that, according to our embedding, $\mathtt{TPP}(Y_1, Y_2)$ implies that $\mathcal{C}(Y_1) \cap \overline{Y_2} \neq \emptyset$ and $Y_1 \subseteq Y_2$; the latter also implies, by monotonicity of closure, $\mathcal{C}(Y_1) \subseteq \mathcal{C}(Y_2)$. Take $Z = \mathcal{C}(Y_1) \cap \overline{Y_2}$. We show that $\mathtt{EC}(Z, Y_1)$ holds, i.e. $\mathtt{C}(Z, Y_1)$ and $\neg\mathtt{O}(Z, Y_1)$: $Z \subseteq \mathcal{C}(Y_1)$ implies[3] $\mathcal{C}(Z) \cap Y_1 \neq \emptyset$; moreover $\mathcal{C}(Y_1) \cap Z = Z$ and $Z \neq \emptyset$ by hypotesis; so $\mathtt{C}(Z, Y_1)$ holds. $Y_1 \subseteq Y_2$ implies $Y_1 \cap \overline{Y_2} = \emptyset$, which in turn implies $Z \cap Y_1 = \emptyset$, i.e. $\neg\mathtt{O}(Z, Y_1)$.

Now we show that $\mathtt{EC}(Z, Y_2)$ holds, i.e. $\mathtt{C}(Z, Y_2)$ and $\neg\mathtt{O}(Z, Y_2)$: We have already proved $\mathcal{C}(Z) \cap Y_1 \neq \emptyset$; so we get $\emptyset \neq \mathcal{C}(Z) \cap Y_1 \subseteq \mathcal{C}(Z) \cap Y_2$ because $Y_1 \subseteq Y_2$, i.e. $\mathcal{C}(Z) \cap Y_2 \neq \emptyset$; moreover $\emptyset \neq \mathcal{C}(Y_1) \cap \overline{Y_2} = \mathcal{C}(Y_1) \cap (\mathcal{C}(Y_1) \cap \overline{Y_2}) \subseteq \mathcal{C}(Y_2) \cap (\mathcal{C}(Y_1) \cap \overline{Y_2})$ because $\mathcal{C}(Y_1) \subseteq \mathcal{C}(Y_2)$ and $\mathcal{C}(Y_2) \cap (\mathcal{C}(Y_1) \cap \overline{Y_2}) = \mathcal{C}(Y_2) \cap Z$; so $\mathtt{C}(Z, Y_2)$ holds. $Z \subseteq \overline{Y_2}$ implies $Z \cap Y_2 = \emptyset$, i.e. $\neg\mathtt{O}(Z, Y_2)$. In conclusion, we proved that there exists Z such that $\mathtt{EC}(Z, Y_1)$ and $\mathtt{EC}(Z, Y_2)$ which, together with $P(Y_1, Y_2)$, completes the first half of the proof.

Now, suppose that $\mathtt{PP}(Y_1, Y_2)$ and there exists Z such that $\mathtt{EC}(Z, Y_1)$ and $\mathtt{EC}(Z, Y_2)$; then $Z \subseteq \overline{Y_2}$, because $\mathtt{EC}(Z, Y_2)$ implies $\neg\mathtt{O}(Z, Y_2)$; moreover, EC is commutative, so we have also $\mathtt{EC}(Y_1, Z)$, which implies $\mathcal{C}(Y_1) \cap Z \neq \emptyset$, and then $\mathcal{C}(Y_1) \cap \overline{Y_2} \neq \emptyset$, since $Z \subseteq \overline{Y_2}$. The above, together with $\mathtt{P}(Y_1, Y_2)$, completes the proof.

Correctness of our definition of NTPP can be proved in a similar way and is left to the reader.

Note that our definition of $[\![\mathtt{C}(Y_1, Y_2)]\!]$ could be simplified to $\neg[\![\mathcal{C}(Y_1) \cap Y_2 = \emptyset]\!]$ due to symmetry of the adjacency relation. We prefer the more general definition covering also the case in which the underlying relation is not symmetric. Finally, our definition of $[\![\mathtt{TPP}(Y_1, Y_2)]\!]$ resembles the alternative definition by equation (32) in [18].

5 Model checking RCC8D using `topochecker`

The tool `topochecker` is a global spatio-temporal model checker, capable of analysing either directed graphs, or digital images. The tool is implemented in the functional programming language OCaml[4], catering for a good balance between declarative features and computational efficiency. The algorithms implemented by `topochecker` are linear in the size of the input space. The spatial model checking algorithm is run in central memory, and it uses memoization and on-disk caching to store intermediate results, achieving high efficiency.

[3] It is trivial to prove that, for quasi-discrete closure space $(X, \mathcal{C}_R)$, whenever R is symmetric, if $B \subseteq \mathcal{C}_R(A)$ then $\mathcal{C}_R(B) \cap A \neq \emptyset$, for all non-empty $A, B \subseteq X$.

[4] See `http://www.ocaml.org`.

For SLCS formulas, the output of the tool consists of a copy of its input, where the points on which each user-defined formula holds are indicated, e.g. by colouring pixels (for images), or labelling nodes (for graphs). Although such mechanism is quite useful (for instance, because it permits one to colour so called "regions of interest" in medical images), it is not apt to report the result of checking CSLCS formulas of the form $\mathcal{M}, \emptyset \models_C \phi$. This is so because the application of Proposition 2 results in a truth value, not a set of points that satisfy the property. In order not to change the way `topochecker` produces its results, and to permit the use of both "truth-valued" CSLCS formulas and "point-valued" SLCS formulas at the same time, the tool has been augmented with a *conditional* formula constructor. Using this constructor, one can define a new point-valued formula Φ by

$$\texttt{Let } \Phi \texttt{ = IF } \Psi \texttt{ THEN } \Phi_1 \texttt{ ELSE } \Phi_2 \texttt{ FI}$$

where Ψ is a CSLCS truth-valued formula, whereas Φ_1 and Φ_2 are SLCS point-valued formulas. The result of such a definition is that Φ is true on the points where Φ_1 holds, if $\mathcal{M}, \emptyset \models_C \Psi$, and on the points where Φ_2 holds, otherwise. Formulas Φ_1 and Φ_2 can for instance be atomic propositions that denote "indicator" areas that make truth of the CSLCS formula Ψ observable as graphical output. Application of the conditional constructor is not limited to image models; for instance, given a quasi-discrete closure-space $(X, \mathcal{C}_R)$, one can augment the space with new *isolated* points (these are by definition not connected to X via R), and new special *indicator* atomic propositions, which characterize each new point. These indicator atomic propositions can then be used to produce output in topochecker via the conditional constructor. The formal details are left as an exercise.

We use this conditional constructor in the example in Figure 7 to illustrate the `TPP` operator using `topochecker`. On the left, an RGB image is displayed,

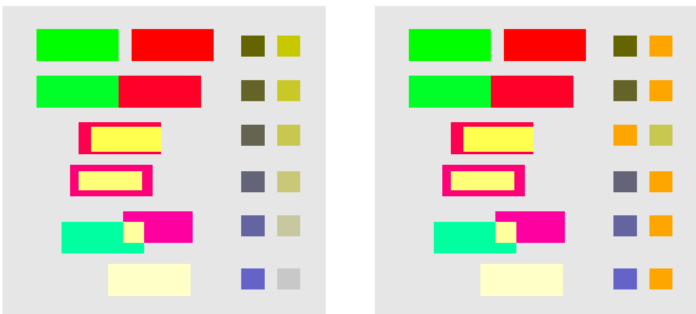


Fig. 7: Checking the `TPP` operator using `topochecker`.

where each pixel has three colour components, red, green and blue, respectively, each ranging over 8 bits (i.e. taking values from 0 to 255). This is the input of the model checking session. Such input image consists of six rows, each one containing a green-ish rectangle g_i on the left, a red-ish rectangle r_i on the right of it (for i in $\{1, \ldots, 6\}$) and two more squares further to the right yes_i (the leftmost square) and no_i (the rightmost square).

All the information is encoded in the red, green and blue components of each pixel. Each of the six rows is identified by a different shade of the blue component of the pixels in that row (for example, in the topmost row the value of the blue component of each pixel is equal to 0, whereas in row 3 the blue component of each pixel is equal to 80); the pixels in each green-ish rectangle have their green component equal to 255, whereas the pixels in each red-ish rectangle have their red component equal to 255 (therefore, when g_i and r_i overlap, the overlapping area has both red and green components equal to 255, that is, it shows up as a yellow-ish area in Fig. 7). Each pixel in a yes_i square has both red and green components that are equal to 100, whereas each pixel in a no_i square has both red and green components that are equal to 200.

Atomic properties for RGB images in `topochecker` are equalities and comparisons on colour components. For instance one can define the points of $g_3 \cup r_3 \cup yes_3 \cup no_3$ by[5]

```
Let row3 = [blue == 80]
```

because all pixels in the third row have their blue component set to 80. The points of $\bigcup_i r_i$, i.e. all red pixels in all rows, can be identified by

```
Let right = [red == 255]
```

therefore, the red rectangle in row 3, namely r_3, is characterised by the formula `row3 & right`.

On the right of Figure 7, the image produced by `topochecker` as a result of spatial model checking is shown. For each row i, one of the two (right-most) squares has been coloured in orange. More precisely, yes_i is coloured if $\texttt{TPP}(g_i, r_i)$ holds, and no_i is coloured otherwise (indeed, only $\texttt{TPP}(g_3, r_3)$ actually holds). Such image is produced by the following statement:

```
Check "orange" checktpp(row1) | checktpp(row2) | checktpp(row3)

        | checktpp(row4) | checktpp(row5) | checktpp(row6)
```

where `checktpp` is a conditional definition that, for each row i, identifies either yes_i or no_i according to the satisfaction value of $\texttt{TPP}(g_i, r_i)$. Since the current version[6] of `topochecker` permits only the definition of point-valued macros (not of truth-valued ones), the encoding of RCC8D in the definition of `checktpp` has been expanded manually, as follows:

[5] In the remainder of this section, we employ the syntax of `topochecker`, using `&` for conjunction, `|` for disjunction, `!` for negation, `-<` for the "share" connective, and `Gr` for the "group" connective.

[6] This may change in a future release of the model checker.

```
Let green(row) = row & left;

Let red(row) = row & right;

Let checktpp(row) =
    IF  ((green(row) & (!red(row))) -< Gr FF) &
        (!((red(row) & (!green(row))) -< Gr FF)) &
        (!(((N green(row)) & (!red(row))) -< Gr FF))
    THEN yes & row
    ELSE no & row
    FI;
```

The condition of the `IF`-statement is a direct encoding of the `TPP` operator into basic CSLCS operators following the encoding defined in Sect. 4. If the condition holds then `TPP`(g_i, r_i) holds in the given row, i.e. the green area is indeed a tangential proper part of the red area, and therefore the small square in the third column is coloured orange, otherwise the square in the fourth, rightmost column is coloured orange. This produces the results in Fig. 7 (right). Of course, this is only one way to visualise the model checking results that exploits the current features of `topochecker` and used here for the purpose of illustration. Other ways can be defined or added as preferred or required by the application at hand.

6 Conclusions

We defined an encoding of the mereotopological theory RCC8D as a fragment of the *Collective Spatial Logic of Closure Spaces* (CSLCS). CSLCS comes equipped with a model checking algorithm and tool, which also contains an experimental spatio-temporal extension of the logic. The newly defined encoding adds a region-based point of view to the point-based methodology of the existing framework. Such developments can be used right away in current applications of spatial and spatio-temporal model checking, including spatio-temporal properties of smart transportation systems [11,14], and medical imaging case studies [2]. Especially for the latter, it is worth mentioning that a new tool is being developed, which is specialised for digital images (including 3D —e.g., magneto-resonance— scans for medical purposes). The tool, called `VoxLogicA` and described in [4], achieves a two-orders-of-magnitude speedup in the specialised setting. `VoxLogicA` does not yet implement the collective operators of CSLCS, but this is a planned development, enabling, by the encoding of RCC8D we propose, efficient image analysis with both the point of views of points and regions. Another interesting domain of application could be that of the characterisation of spatial properties and relations in the context of simulation of biological systems [8,7,6].

One open question regards RCC8D interpreted in arbitrary closure spaces, not just the symmetric ones. We consider worth investigating in future work what operators may be obtained when the underlying relation is directed. Indeed, many relations can be defined (for instance, region A may be "half-connected" to region B when there is an edge from A to B even if there is no edge from B to A). Application domains and case studies will help to clarify which ones make more sense in practice.

7 Acknowledgements

This paper was written for the Festschrift in honour of Prof. Rocco De Nicola. We would like to thank Rocco for the many years of fruitful collaboration in the context of numerous European and Italian research projects and we are looking forward to future collaboration in the context of the new Italian MIUR PRIN project "IT MATTERS". But most of all, we are grateful for his great sense of humanity with which he dedicated part of his professional live to keep computer science research alive in areas struck by devastating earthquakes and to give a second professional chance to people from conflict areas.

References

1. Aiello, M., Pratt-Hartmann, I., van Benthem, J. (eds.): Handbook of Spatial Logics. Springer (2007)
2. Banci Buonamici, F., Belmonte, G., Ciancia, V., Latella, D., Massink, M.: Spatial Logics and Model Checking for Medical Imaging. International Journal on Software Tools for Technology Transfer. Springer. (2019), `https://doi.org/10.1007/s10009-019-00511-9`
3. Belmonte, G., Ciancia, V., Latella, D., Massink, M.: From collective adaptive systems to human centric computation and back: Spatial model checking for medical imaging. In: ter Beek, M.H., Loreti, M. (eds.) Proceedings of the Workshop on FORmal methods for the quantitative Evaluation of Collective Adaptive SysTems, FORECAST@STAF 2016, Vienna, Austria, 8 July 2016. EPTCS, vol. 217, pp. 81–92 (2016), `https://doi.org/10.4204/EPTCS.217.10`
4. Belmonte, G., Ciancia, V., Latella, D., Massink, M.: Voxlogica: a spatial model checker for declarative image analysis. In: Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2019) - Part I. Lecture Notes in Computer Science, vol. 11427. Springer International Publishing (2019), to appear (preprint available at `http://arxiv.org/abs/1811.05677`)
5. Bennett, B., Düntsch, I.: Axioms, algebras and topology. In: Springer [1], pp. 99–159
6. Binchi, J., Merelli, E., Rucco, M., Petri, G., Vaccarino, F.: jholes: A tool for understanding biological complex networks via clique weight rank persistent homology. Electr. Notes Theor. Comput. Sci. 306, 5–18 (2014), `https://doi.org/10.1016/j.entcs.2014.06.011`
7. Buti, F., Cacciagrano, D., Corradini, F., Merelli, E., Tesei, L., Pani, M.: Bone remodelling in bioshape. Electr. Notes Theor. Comput. Sci. 268, 17–29 (2010), `https://doi.org/10.1016/j.entcs.2010.12.003`
8. Buti, F., Cacciagrano, D., Donato, M.C.D., Corradini, F., Merelli, E., Tesei, L.: Bioshape\textsc{BioShape}: End-user development for simulating biological systems. In: Costabile, M.F., Dittrich, Y., Fischer, G., Piccinno, A. (eds.) End-User Development - Third International Symposium, IS-EUD 2011, Torre Canne (BR),

Italy, June 7-10, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6654, pp. 379–382. Springer (2011), https://doi.org/10.1007/978-3-642-21530-8_45
9. Ciancia, V., Latella, D., Loreti, M., Massink, M.: Specifying and verifying properties of space. In: Theoretical Computer Science - 8th IFIP TC 1/WG 2.2 International Conference, TCS 2014, Rome, Italy, September 1-3, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8705, pp. 222–235. Springer (2014)
10. Ciancia, V., Latella, D., Loreti, M., Massink, M.: Model Checking Spatial Logics for Closure Spaces. Logical Methods in Computer Science Volume 12, Issue 4 (Oct 2016), http://lmcs.episciences.org/2067
11. Ciancia, V., Gilmore, S., Grilletti, G., Latella, D., Loreti, M., Massink, M.: Spatio-temporal model checking of vehicular movement in public transport systems. STTT 20(3), 289–311 (2018), https://doi.org/10.1007/s10009-018-0483-8
12. Ciancia, V., Latella, D., Loreti, M., Massink, M.: Spatial logic and spatial model checking for closure spaces. In: Bernardo, M., Nicola, R.D., Hillston, J. (eds.) Formal Methods for the Quantitative Evaluation of Collective Adaptive Systems - 16th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2016, Bertinoro, Italy, June 20-24, 2016, Advanced Lectures. Lecture Notes in Computer Science, vol. 9700, pp. 156–201. Springer (2016), https://doi.org/10.1007/978-3-319-34096-8_6
13. Ciancia, V., Latella, D., Massink, M., Paskauskas, R.: Exploring spatio-temporal properties of bike-sharing systems. In: 2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASO Workshops 2015, Cambridge, MA, USA, September 21-25, 2015. pp. 74–79. IEEE Computer Society (2015), https://doi.org/10.1109/SASOW.2015.17
14. Ciancia, V., Latella, D., Massink, M., Paskauskas, R., Vandin, A.: A tool-chain for statistical spatio-temporal model checking of bike sharing systems. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October 10-14, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9952, pp. 657–673 (2016), https://doi.org/10.1007/978-3-319-47166-2_46
15. Galton, A.: The mereotopology of discrete space. In: Freksa, C., David, M. (eds.) Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science, Lecture Notes in Computer Science, vol. 1661, pp. 251–266. Springer Berlin Heidelberg (1999), http://dx.doi.org/10.1007/3-540-48384-5_17
16. Galton, A.: A generalized topological view of motion in discrete space. Theor. Comput. Sci. 305(1-3), 111–134 (2003), https://doi.org/10.1016/S0304-3975(02)00701-6
17. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Nebel, B., Rich, C., Swartout, W.R. (eds.) Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92). Cambridge, MA, USA, October 25-29, 1992. pp. 165–176. Morgan Kaufmann (1992)
18. Randell, D.A., Landini, G., Galton, A.: Discrete mereotopology for spatial reasoning in automated histological image analysis. IEEE Trans. Pattern Anal. Mach. Intell. 35(3), 568–581 (2013), https://doi.org/10.1109/TPAMI.2012.128
19. Smyth, M.B., Webster, J.: Discrete spatial models. In: Springer [1], pp. 713–798
20. van Benthem, J., Bezhanishvili, G.: Modal logics of space. In: Handbook of Spatial Logics, pp. 217–298. Springer (2007)

A Proof of Proposition 1

Proof. We prove that, for all models $\mathcal{M} = ((X, \mathcal{C}), \mathcal{V})$ and points $x \in X$, the following holds:

$$\mathcal{M}, x \not\models \Phi_2 \vee (\Phi_1 \, \mathcal{S} \, \Phi_2) \text{ iff } \mathcal{M}, x \not\models A(\Phi_1 \, \mathcal{W} \, \Phi_2).$$

For the direct implication, we proceed as follows:

$\mathcal{M}, x \not\models \Phi_2 \vee (\Phi_1 \, \mathcal{S} \, \Phi_2)$

$\Rightarrow$ {Logic}

$\mathcal{M}, x \not\models \Phi_2$ and $\mathcal{M}, x \not\models \Phi_1 \, \mathcal{S} \, \Phi_2$

$\Rightarrow$ {def. of $\mathcal{S}$}

$\mathcal{M}, x \not\models \Phi_2$ and

there exists π, ℓ s.t.

$\pi(0) = x, \mathcal{M}, \pi(\ell) \not\models \Phi_1,$ and $\mathcal{M}, \pi(j) \not\models \Phi_2,$ for all j s.t. $0 < j \leq \ell$

$\Rightarrow$ {Logic}

there exists π, ℓ s.t.

$\pi(0) = x, \mathcal{M}, \pi(\ell) \not\models \Phi_1,$ and $\mathcal{M}, \pi(j) \not\models \Phi_2,$ for all j s.t. $0 \leq j \leq \ell$

$\Rightarrow$ {def. of $\mathcal{W}$}

$\mathcal{M}, \pi \not\models \Phi_1 \, \mathcal{W} \, \Phi_2$

$\Rightarrow$ {def. of A}

$\mathcal{M}, x \not\models A(\Phi_1 \, \mathcal{W} \, \Phi_2)$

For the one but last step of the above derivation, note that: (i) $\mathcal{M}, \pi(\ell) \not\models \Phi_1$ implies that $\mathcal{M}, \pi(i) \models \Phi_1$ for all i does *not* hold; and (ii) $\mathcal{M}, \pi(j) \not\models \Phi_2$, for all j s.t. $0 \leq j \leq \ell$ implies that, if there exists k s.t. $\mathcal{M}, \pi(k) \models \Phi_2$, then, it necessarily must be $k > \ell$; but then $\mathcal{M}, \pi \models \Phi_1 \, \mathcal{U} \, \Phi_2$ cannot hold because this would *not* allow $\mathcal{M}, \pi(\ell) \not\models \Phi_1$, with $\ell < k$.

The derivation for the reverse implication is given below:

$\mathcal{M}, x \not\models A(\Phi_1 \, \mathcal{W} \, \Phi_2)$

$\Rightarrow$ {def. of A}

there exists π s.t.

$\pi(0) = x$ and $\mathcal{M}, \pi \not\models \Phi_1 \, \mathcal{W} \, \Phi_2$

$\Rightarrow$ {def. of $\mathcal{W}$}

there exist π, ℓ s.t.

$\pi(0) = x$ and $\mathcal{M}, \pi(\ell) \not\models \Phi_1$ and $\mathcal{M}, \pi \not\models \Phi_1 \,\mathcal{U}\, \Phi_2$

$\Rightarrow$ $\{\mathcal{M}, \pi \not\models \Phi_1 \,\mathcal{U}\, \Phi_2$ implies $\mathcal{M}, \pi(0) \not\models \Phi_2\}$

there exists π, ℓ s.t.

$\pi(0) = x$ and $\mathcal{M}, \pi(\ell) \not\models \Phi_1$ and $\mathcal{M}, \pi(0) \not\models \Phi_2$ and $\mathcal{M}, \pi \not\models \Phi_1 \,\mathcal{U}\, \Phi_2$

Take the minimal ℓ as above. If $\ell = 0$, then clearly $\mathcal{M}, x \not\models \Phi_1 \mathcal{S} \Phi_2$, by definition of $\mathcal{S}$, and since we also have $\mathcal{M}, x \not\models \Phi_2$, we get $\mathcal{M}, x \not\models \Phi_2 \vee (\Phi_1 \mathcal{S} \Phi_2)$, i.e. the assert. If instead $\ell > 0$, then clearly $\mathcal{M}, \pi(j) \models \Phi_1$ for $0 \leq j < \ell$, by minimality of ℓ, and since we also have $\mathcal{M}, \pi \not\models \Phi_1 \,\mathcal{U}\, \Phi_2$, we get $\mathcal{M}, \pi(j) \not\models \Phi_2$ for $0 \leq j \leq \ell$. So, there exist π, ℓ s.t. $\pi(0) = x$, $\mathcal{M}, \pi(\ell) \models \neg\Phi_1$ and for all j, $0 < j \leq \ell$, $\mathcal{M}, \pi(j) \not\models \Phi_2$, which, by definition of $\mathcal{S}$, is equivalent to $\mathcal{M}, x \not\models \Phi_1 \,\mathcal{S} \Phi_2$. Moreover, since we also know that $\mathcal{M}, \pi(0) \not\models \Phi_2$, we get $\mathcal{M}, x \not\models \Phi_2 \vee (\Phi_1 \mathcal{S} \Phi_2)$, i.e. the assert.