Geometric Model Checking of Continuous Space *

Nick Bezhanishvili[1], Vincenzo Ciancia[2][0000−0003−1314−0574], David Gabelaia[3][0000−0002−8317−7949], Gianluca Grilletti[1][0000−0002−1631−3648], Diego Latella[2][0000−0002−3257−9059], and Mieke Massink[2][0000−0001−5089−002X]

[1] Institute for Logic, Language and Computation, University of Amsterdam
[2] Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", Consiglio Nazionale delle Ricerche
[3] TSU Razmadze Mathematical Institute, Tbilisi, Georgia

Abstract. Topological Spatial Model Checking is a recent paradigm that combines Model Checking with the topological interpretation of Modal Logic. The Spatial Logic of Closure Spaces, SLCS, extends Modal Logic with reachability connectives that, in turn, can be used for expressing interesting spatial properties, such as "being near to" or "being surrounded by". SLCS constitutes the kernel of a solid logical framework for reasoning about *discrete* space, such as graphs and digital images, interpreted as quasi discrete closure spaces. In particular, the spatial model checker VoxLogicA, that uses an extended version of SLCS, has been used successfully in the domain of medical imaging. However, SLCS is not restricted to discrete space. Following a recently developed *geometric* semantics of Modal Logic, we show that it is possible to assign an interpretation to SLCS in *continuous space*, admitting a model checking procedure, by resorting to models based on polyhedra. In medical imaging such representations of space are increasingly relevant, due to recent developments of 3D scanning and visualisation techniques that exploit mesh processing. We demonstrate feasibility of our approach via a new tool, PolyLogicA, aimed at efficient verification of SLCS formulas on polyhedra, while inheriting some well-established optimization techniques already adopted in VoxLogicA. Finally, we cater for a geometric definition of bisimilarity, proving that it characterises logical equivalence.

Keywords: Spatial Logic · Model Checking · Geometric Logic

1 Introduction and Related Work

Recently, novel variants of model checking have been developed, moving the focus from checking *temporal* properties to *spatial* properties, see for example [33,20,21,24,41,34], and, in fact, also to the combination of reasoning on time and space in spatio-temporal model checking [18,25,19,23,22,31,43].

* Research partially supported by the MIUR Project PRIN 2017FTXR7S "IT- MaT-TerS". The authors are listed in alphabetical order, as they equally contributed to this work.

The so-called *topological* approach to spatial logic and spatial model checking has its origin in the ideas by McKinsey and Tarski, who recognised the possibility of reasoning about space using topology as a mathematical framework for the interpretation of modal logic (see [3] for a thorough introduction). The work by Ciancia et al. (see e.g. [20,21]) builds on these theoretical developments using *Closure Spaces*, a generalisation of topological spaces encompassing also general *discrete* spatial structures such as graphs [28,29], as underlying model for the *Spatial Logic for Closure Spaces* (SLCS). The original version of this spatial logic included two spatial operators, the *near* operator and the *surrounded* operator. Points in space satisfying 'near ϕ' are all those points close to any point satisfying ϕ. Points satisfying 'ϕ surrounded by ψ', instead, are all those points satisfying ϕ from which no path can be found that passes by a point not satisfying ϕ without first passing by a point satisfying ψ. In other words, these are those points, satisfying ϕ, that are surrounded by points satisfying ψ.

Two different spatial model-checkers for finite (quasi-discrete) closure spaces were developed based on this foundational work: Topochecker and VoxLogicA. These tools have been used in several application areas, ranging from collective adaptive systems [25,23,22] to signals [41] and medical image analysis [33]. Recently, in this latter domain of application, promising results have been obtained for the segmentation of malignant brain lesions [10] and normal brain tissue such as white and grey matter [9], as well as for segmentation of nevi[4] [8]. Note that in medical image analysis, model checking addresses *static* properties of space instead of *dynamic* properties of agents moving through space as in [23].

So far, spatial model checking focused on *discrete* spatial structures, seen as discrete sets of points, i.e. nodes of graphs. Here, instead, we focus on the development of the foundations to reason about and model-check *continuous* space. This is certainly motivated by the theoretical foundations of the research line (after all, closure spaces comprise both continuous and discrete space) but also by the developments in the domain of medical image analysis and visualisation. In particular, computerised 3D visualisation of medical images can help physicians to make better diagnoses or treatment plans. Images used for visualisation often consist of *continuous* spatial structures that are divided into suitable areas of different size using mesh techniques such as *triangular surface meshes* or *tetrahedral volume meshes* (see for example [38]). In order to develop a formalism capable of feasible model-checking on such structures, we build upon recent developments in polyhedral semantics for modal and intuitionistic logic [12,2,1]. Unlike the topological semantics, where formulas are interpreted in the powerset algebra of a topological space, in this semantics formulas are assigned polyhedral subsets of an n-dimensional Euclidean space. Polyhedral subsets can be thought of as finite unions of simplexes (n-dimensional triangles). Using piecewise linear geometry (triangulations, nerves), [12] gives a full characterization of the intuitionistic and modal logics of the class of all compact polyhedra and [2,1] provide an infinite family of polyhedrally complete modal and intermediate logics.

[4] Benign and malignant lesions of the skin.

In this paper we take the polyhedral semantics one step further by extending the language with a *spatial reachability* modality, namely a variant the ρ operator originally proposed in [10]. The reachability modality γ we use in the present paper is a binary logical operator that can be seen as a variant of the spatial interpretation of the Existential Until temporal operator. Roughly speaking, $\gamma(\phi, \psi)$ (pronounced as ψ is reachable through ϕ) means that a point satisfying ψ is reachable by a path satisfying ϕ along the way. The reachability modality is quite expressive and other operators, relevant for the intended applications (such as "surrounded", or "grow", discussed in more detail throughout the paper), can be defined using it. We show that the reachability modality can be defined in polyhedral models. We define a notion of *bisimilarity* between two polyhedral models, and we prove that bisimilarity preserves and reflects logical equivalence. Moreover, we prove that the continuous model of the extended language can be turned into a finite relational model for the same language without losing any of the logical information.

In fact, one of the main conceptual results of the present paper is that a formula ϕ is satisfiable in a polyhedral model $\mathcal{X}$ iff ϕ is satisfiable in a relational (Kripke) model $\mathcal{M}(\mathcal{X})$ obtained from $\mathcal{X}$ in a uniform way. In particular, $\mathcal{M}(\mathcal{X})$ is the face poset of an underlying triangulation of $\mathcal{X}$. Triangulation is a standard technique of piecewise linear geometry letting one to approximate each polyhedron via simplexes. That triangulations play an important role in logical analysis of polyhedra has already been observed in [12,1,2]. However, here we demonstrate this also for the language enriched with the reachability modality γ. Thus, $\mathcal{M}(\mathcal{X})$ provides a full logical invariant for $\mathcal{X}$.

The finite state, Kripke-style semantics that we define preserves all the information that can be discerned by SLCS formulas. This is the key for introducing a novel *geometric model checking* technique to analyse continuous space. A model checking algorithm, along the lines of [10], has been implemented in the free and open source geometric model checker PolyLogicA, which brings to the continuous domain the core features of VoxLogicA (global model checking, concurrent multi-core execution, "memoization" at the syntactic level).

Further related work. The theoretical framework for spatial model checking of continuous space in the present paper is based on spatial models involving polyhedra. Polyhedra also play an important role in development of model checking algorithms for the verification of behavioural properties of real-time and hybrid systems (see for example [36,5,13,35,6] and references therein). In that context polyhedra, and their related notions such as template polyhedra [42,13] and zonotopes [30], are obtained from sets of linear equations involving real-time constraints on system behaviour and are a natural representation of sets of states of such systems. In the present paper we focus on *spatial* properties of continuous space rather than behavioural properties.

Outline. Section 2 introduces the basic geometrical notions and notation. Section 3 recalls SLCS and provides its semantics on polyhedral models. In Section 4

the concept of simplicial bisimilarity is introduced and it is shown that it characterises logic equivalence for SLCS formulas. Section 5 and Section 6 present the foundations for geometric model checking and the related model checker PolyLogicA, resp. Section 7 concludes the paper with an outlook for future work.

Proofs of most relevant facts are reported in Appendix A.

2 Background

In this section, we establish the basic geometric notions that we use in this work. See [40, Chapter 2] for more details on these matters.

Definition 1 (Simplex). *An* n-simplex σ *is the convex hull of a finite set* $V = \{v_0, v_1, \ldots, v_n\} \subseteq \mathbb{R}^d$ *of affinely independent points*[5]*, that is the set* $\sigma = \{\lambda_0 v_0 + \cdots + \lambda_n v_n \mid \forall i.\lambda_i \in [0,1] \text{ and } \sum_{i=0}^{n} \lambda_i = 1\}$*. The number* n *is called the* dimension *of* σ *and* $v_0, \ldots, v_n$ *are called its* vertices.

In Definition 1, any subset of $\{v_0, \ldots, v_n\}$ is also a set of affinely independent points, and thus it spans a simplex τ: we call τ a *face* of σ (in symbols $\tau \preceq \sigma$), and we call it a *proper face* if $\tau \neq \emptyset$ and $\tau \neq \sigma$. Next, we identify the "internal part" of a simplex.

Definition 2 (Relative interior). *In the notations of Definition 1, let the* relative interior *of* σ *be the set* $\widetilde{\sigma} := \{\sum_{i=0}^{n} \lambda_i v_i \mid \forall i.\lambda_i \in (0,1] \text{ and } \sum_{i=0}^{n} \lambda_i = 1\}$.

Note that if σ is non-empty then also $\widetilde{\sigma}$ is non-empty. For instance, $b_\sigma := \sum_{i=0}^{n} \frac{1}{n+1} v_i$ (i.e., the *barycentre* of σ) is an element of $\widetilde{\sigma}$. In particular, the relative interior of a point p is p itself. Each simplex σ is partitioned by the relative interiors of its faces, that is, $\sigma = \bigcup\{\widetilde{\tau} \mid \tau \preceq \sigma\}$.

Given a topological space, let us indicate with $\mathcal{C}$ and $\mathcal{I}$ the closure and interior operations respectively. Being subsets of an Euclidean space $\mathbb{R}^m$, simplexes inherit the topology from the "ambient space". In particular, each simplex σ is the topological closure of $\widetilde{\sigma}$, that is, $\sigma = \mathcal{C}(\widetilde{\sigma})$. More complex spaces are obtained by "gluing together" simplexes.

Definition 3 (Simplicial complex). *A* simplicial complex K *is a finite set of simplexes of* $\mathbb{R}^m$ *such that:*

1. *If* $\sigma \in \mathrm{K}$ *and* τ *is a face of* σ*, then* $\tau \in \mathrm{K}$*;*
2. *If* $\sigma, \tau \in \mathrm{K}$*, then* $\sigma \cap \tau$ *is a face of* σ *and* τ *(eventually the empty simplex).*

The *dimension* of K is the maximum of the dimensions of its simplexes. The set of points that lie in at least one of the simplexes of K is a topological space called the *polyhedron* of K, formally defined as $|\mathrm{K}| := \bigcup \mathrm{K}$. A point of $|\mathrm{K}|$ may belong to several of the simplexes in K. However, there is a natural way to associate to each point of $|\mathrm{K}|$ the "smallest" simplex it belongs to.

[5] $v_0, \ldots, v_n$ are *affinely independent* if $v_1 - v_0, \ldots, v_n - v_0$ are linearly independent. In particular, this condition implies that $n \leq d$.

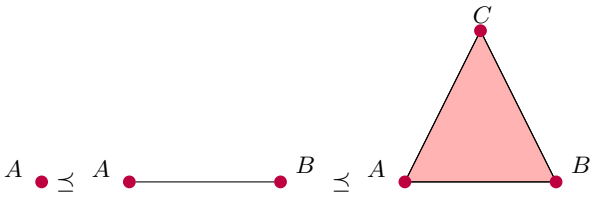


Fig. 1. An example of simplicial complex (the rightmost triangle), and some of the faces that compose it in the $\preceq$ relation.

Lemma 4. *Each point of* $|\mathrm{K}|$ *belongs to the relative interior of exactly one non-empty simplex in* K. *That is,* $\widetilde{\mathrm{K}} := \{\widetilde{\sigma} \mid \sigma \in \mathrm{K} \setminus \{\emptyset\}\}$ *is a partition of* $|\mathrm{K}|$.

We call $\widetilde{\mathrm{K}}$ a *simplicial partition* of $|K|$, and we call its elements the *cells* of the partition.[6] Note that distinct simplicial complexes induce distinct partitions, even when they are associated to the same polyhedron. From now on, to ease readability, we fix a simplicial complex K, with the associated $|\mathrm{K}|$ and $\widetilde{\mathrm{K}}$.
Finally, we recall the topological notion of *path*.

Definition 5. *A topological* path *is a total, continuous function* $\pi : [0,1] \to P$, *where* $[0,1]$ *is equipped with the subspace topology of* $\mathbb{R}$.

With a mild abuse of notation, for S a subset of $[0,1]$ and π a path, we write $\pi(S)$ to denote $\{\pi(x) \mid x \in S\}$.

3 Interpreting SLCS on Polyhedra

In this section we introduce the main theory driving our model checking approach to polyhedra. In the classical topological tradition, valuations of atomic propositions can be arbitrary subsets of the space. In this work, instead, we restrict our attention only to a specific class of spatial regions, namely union of cells of a fixed simplicial partition. This simple change permits us to define SLCS on continuous space, while retaining decidability of the model checking problem. First of all, we introduce the syntax of the variant of SLCS that we use in this paper, that is based on the binary modality γ instead of ρ of [10]; the relationship between ρ and γ will be shown in Proposition A.1.

Definition 6 (Syntax). *The syntax of the logic SLCS is:*

$$\phi \ ::= \ \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \Box\phi \mid \gamma(\phi, \phi)$$

where p *is an atomic proposition, taken from a fixed finite set* AP.

[6] We use the terminology *cells* in this way for the purposes of this paper; there is no relation between such cells and the so-called *cell complexes* of algebraic topology.

Thus, we enhance the basic modal language with a *reachability* operator γ. As in the standard topological semantics for modal logic, we interpret formulas as sets of points. Boolean operators (disjunction $\vee$ and negation $\bot$ are derived via De Morgan laws) are given their standard set-theoretical interpretation. The $\Box$ modality corresponds to topological interior. The formula $\gamma(\phi, \psi)$ ("reach ψ through ϕ") is satisfied by a point if there is a path rooted at that point, leading to a point satisfying ψ and whose intermediate points all satisfy ϕ.

Next, we shall introduce the models and semantics of our logic. We indicate with $\mathcal{P}(P)$ the powerset of P.

Definition 7 (Model). *A* Polyhedral Model *is a triplet* $\mathcal{X} = \langle P, \mathrm{K}, V \rangle$*, where* $P \subseteq \mathbb{R}^d$ *is a polyhedron,* K *is a simplicial complex such that* $P = |\mathrm{K}|$*, and* $V : \mathrm{AP} \to \mathcal{P}(P)$ *is a valuation such that* $V(p)$ *is a union of cells of* $\widetilde{\mathrm{K}}$*.*

Polyhedral models are essentially topological models with some extra restrictions on the valuation: P plays the role of the topological space (with the topology induced by the ambient space $\mathbb{R}^d$) and V is used to interpret atomic propositions as specific subsets of this space, namely those that are the union of a finite number of simplicial cells. From now on, fix a polyhedral model $\mathcal{X} = \langle P, \mathrm{K}, V \rangle$.

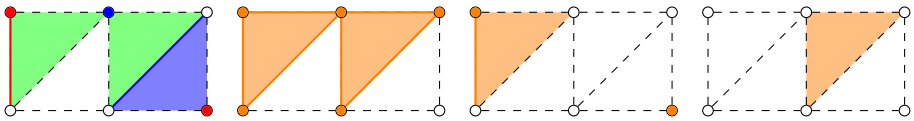
Definition 8 (Semantics). *Given* $x \in P$*, satisfaction* $\mathcal{X}, x \vDash \phi$ *over formulas* ϕ *is given by the following inductive clauses, where we let* $[\![\phi]\!]^{\mathcal{X}}$ *denote the set* $\{x \in P \mid \mathcal{X}, x \vDash \phi\}$*:*

$$
\begin{array}{lll}
\mathcal{X}, x \vDash \top & & \textit{always holds} \\
\mathcal{X}, x \vDash p & \iff & x \in V(p) \textit{ for } p \in \mathrm{AP} \\
\mathcal{X}, x \vDash \neg\phi & \iff & \mathcal{X}, x \nvDash \phi \\
\mathcal{X}, x \vDash \phi \wedge \psi & \iff & \mathcal{X}, x \vDash \phi \textit{ and } \mathcal{X}, x \vDash \psi \\
\mathcal{X}, x \vDash \Box\phi & \iff & x \in \mathcal{I}([\![\phi]\!]^{\mathcal{X}}) \\
\mathcal{X}, x \vDash \gamma(\phi, \psi) & \iff & \textit{there exists a path } \pi \textit{ such that} \\
 & & \pi(0) = x,\ \pi(1) \in [\![\psi]\!]^{\mathcal{X}} \textit{ and } \pi((0,1)) \subseteq [\![\phi]\!]^{\mathcal{X}}
\end{array}
$$

The definition of the satisfaction relation for the standard operators of modal logic is the usual one for the classical topological interpretation. In particular, note the *topological interior* interpretation of $\Box\phi$, intuitively expressing that point x is in the "internal" part of the set of points satisfying ϕ. Regarding reachability, a point x satisfies $\gamma(\phi, \psi)$ in model $\mathcal{X}$ if there is a path π rooted in x leading to a point y satisfying ψ; in addition, all the points that lay in π, except x and y, are required to satisfy ϕ. Indeed, several different variants of reachability could be defined using this operator.

As a prominent example, the reachability modality $\rho\,\psi\,[\phi]$ introduced in [10], that we also employ to introduce some derived operators, can be defined as $\rho\,\psi[\phi] := \psi \vee \gamma(\phi, \psi)$. Actually, the two operators are inter-definable, by letting $\gamma(\phi, \psi) := \rho\,(\phi \wedge \rho\,\psi[\phi])[\phi]$ (see Proposition A.1). In this work we opt to use γ since, in the context of polyhedral models, its definition is more concise.

Another relevant spatial modality is the *surrounded* operator $\mathcal{S}$ (e.g., [21,39,43,41] use it as a primitive of the language). A point x satisfies $\phi\,\mathcal{S}\,\psi$ if it lays in an



(a) Model (b) $\gamma(g, b)$ (c) $grow(r, g)$ (d) $g \wedge \gamma(g, \gamma(b, r))$

Fig. 2. Examples of SLCS formulas on Polyhedra. 2a) Polyhedral model. Circles denote points (1-dimensional simplexes). The valuation of atomic propositions r, g, b is given by the colours *red*, *green*, and *blue*. Dashed segments and white points and triangles do not satisfy any atomic proposition. 2b) Points (in orange) satisfying $\gamma(g, b)$ applied to the model in 2a. Note how some white points in 2a also satisfy such proposition. 2c) Points (in orange) satisfying $grow(r, g)$. Note that only the points corresponding to the red area and one green triangle in the model in 2a are coloured, and no white point. 2d) Points (in orange) satisfying $g \wedge \gamma(g, \gamma(b, r))$; composing reachability, quite complex formulas may be defined.

area whose points satisfy ϕ, and that is limited (i.e., surrounded) by points that satisfy ψ. In other words, it is not possible to exit this area without passing by a point satisfying ψ. Following [10], we can define the operator $\mathcal{S}$ on polyhedral models in terms of ρ through the following expression: $\phi \wedge \neg\rho\,(\neg(\phi \vee \psi))[\neg\psi]$.

Some examples are shown in Figure 3. We refer to the caption of that figure for more detailed explanation. Notably, we illustrate the derived operator $grow(a, b)$, that also played an important role in the brain tumour segmentation procedure presented in [10]. The operator *grow* is reminiscent of the technique of *region growing* in Medical Imaging, and it is used to characterise those areas of space satisfying b that are in contact with areas of space satisfying a, or, in other words, the operator lets a "grow" inside b (and no further). The formal definition is $grow(\phi_1, \phi_2) := \phi_1 \vee touch(\phi_2, \phi_1)$, where $touch(\phi_3, \phi_4) := \phi_3 \wedge \rho\,\phi_4[\phi_3]$.

Note that the same polyhedron P can be associated with different simplicial complexes: our semantics is not sensitive to such presentational ambiguity in the description of P.[7] This is because, although we need to specify K to spell out the restriction on the range of V, K itself does not play a role in the semantics, as shown in the following proposition.

Lemma 9. *Let $\mathcal{X} = \langle P, \mathrm{K}, V\rangle$ and $\mathcal{X}' = \langle P, \mathrm{K}', V\rangle$ two models sharing the same P and V. For each $x \in P$ and ϕ we have: $\mathcal{X}, x \vDash \phi \iff \mathcal{X}', x \vDash \phi$.*

Therefore, for the sake of readability, we will sometimes indicate a polyhedral model with the notation $\mathcal{X} = \langle P, V\rangle$, abstracting from the particular choice of K. Nevertheless, we require V to range over unions of cells of *some* polyhedral partition, thus restricting the semantics to spatial regions definable in terms of polyhedra. We will call a simplicial complex K as in Definition 7 *coherent* with the model $\mathcal{X} = \langle P, V\rangle$.

[7] Such ambiguity can be thought of as being similar, in spirit, to the infinitely many different programs that may result in equivalent Kripke frames, in a specification language for classical model checking applications.

We mentioned that employing polyhedra allows for a *finitary treatment* of the semantics. The following results are essential to formalize this intuition, which will be further investigated in Section 5.

Definition 10. *Let $\mathcal{X} = \langle P, V \rangle$ be a polyhedral model.* Logical equivalence $\equiv$ *is the binary relation on P such that $x \equiv y$ if and only if, for every formula ϕ: $\mathcal{X}, x \vDash \phi \iff \mathcal{X}, y \vDash \phi$.*

Lemma 11. *Let $\mathcal{X}$ be a polyhedral model and* K *a simplicial complex coherent with $\mathcal{X}$. Then for each cell $\widetilde{\sigma} \in \widetilde{\mathrm{K}}$ and $x, y \in \widetilde{\sigma}$ we have $x \equiv y$.*

In particular, for every formula ϕ, $[\![\phi]\!]^{\mathcal{X}}$ is a (finite) union of cells of $\widetilde{\mathrm{K}}$.

Proposition 12. *Given a polyhedral model $\mathcal{X}$, the relation $\equiv$ has only finitely many equivalence classes. Furthermore, each equivalence class C has a* characteristic formula *ϕ^C such that $\mathcal{X}, x \vDash \phi^C \iff x \in C$.*

The above facts are also useful to prove an interesting feature of polyhedral models, namely that the $\Box$ modality can be considered a derived operator, as it expressible using γ. This considerably simplifies proofs.

Theorem 13. *For each formula ϕ, we have $\mathcal{X}, x \vDash \Box\phi \iff \mathcal{X}, x \vDash \neg\gamma(\neg\phi, \top)$.*

Another property of polyhedral models which turns out to be fundamental in this work is that we can restrict our attention to a special class of paths—rather than arbitrary paths—to study the reachability operator γ: *piecewise linear paths*.

Definition 14 (PL-path). *We call a path $\pi : [0,1] \to P$* piecewise linear *(or simply PL-path) if there exist values $r_0 = 0, r_1, \ldots, r_k = 1$ such that for every $i = 0, \ldots, k-1$ and $t \in [0,1]$: $\pi\big(tr_i + (1-t)r_{i+1}\big) = t\pi(r_i) + (1-t)\pi(r_{i+1})$.*

We indicate that a path is piecewise linear with the notation $\pi : [0,1] \overset{PL}{\to} P$. Intuitively, a PL-path is obtained by connecting a finite number of segments and parametrizing them in a suitable way. Even if PL-paths are much simpler than arbitrary paths, when it comes to connectivity in polyhedral models the two classes are interchangeable, as shown in the following lemma. We call a set X *PL-connected* if for all $x, y \in X$ there exists a PL-path from x to y.

Lemma 15. *Let* K *be a simplicial complex and $\Pi \subseteq \widetilde{\mathrm{K}}$. $\bigcup \Pi$ is connected iff $\bigcup \Pi$ is PL-connected.*

Using the previous result, we can give an alternative semantic characterization of the reachability operator, which is relevant for the proofs of decidability of model checking (Section 5) and the characterisation of logical equivalence via bisimilarity (Section 4).

Lemma 16. *We have: $\mathcal{X}, x \vDash \gamma(\phi, \psi)$ if and only if there is a piecewise linear path $\pi : [0,1] \overset{PL}{\to} P$ such that $\pi(0) = x$ and $\pi((0,1)) \subseteq [\![\phi]\!]^{\mathcal{X}}$ and $\pi(1) \in [\![\psi]\!]^{\mathcal{X}}$.*

4 Simplicial Bisimilarity

In this section we characterise logical equivalence via *bisimilarity*. Recall the results summarised in [4], defining bisimilarity for topological spatial logics, so that any two points are bisimilar if and only if they are logically equivalent. Bisimilarity is a fundamental tool in modal logic (see e.g. [44]).
In order to characterise reachability, the definition of bisimilarity requires a special class of paths defined below. In the following, fix a model $\mathcal{X} := \langle P, \mathrm{K}, V \rangle$.

Definition 17. *A path $\pi : [0,1] \to P$ is* simplicial *if and only if there is a finite sequence $s_0 = 0 < \ldots < s_k = 1$ of values in $[0,1]$ and cells $\widetilde{\sigma}_1, \ldots, \widetilde{\sigma}_k \in \widetilde{\mathrm{K}}$ such that, for all $i = 1, \ldots, k$, we have $\pi((s_{i-1}, s_i)) \subseteq \widetilde{\sigma}_i$.*

Notice that the property of being *simplicial* depends on the simplicial complex K. However, we already encountered a family of paths which are simplicial independently from the choice of K.

Lemma 18. *Any piecewise linear path is simplicial.*

The definition of bisimilarity makes use of the point-wise lifting of a relation to a path, defined in a formal way below.

Definition 19. *Given a relation $R \subseteq P \times P$, let the* extension of R to paths *be the binary relation between paths $\hat{R}$, such that $\pi_1 \hat{R} \pi_2$ if and only for all $t \in [0,1]$ we have $\pi_1(t) R \pi_2(t)$.*

Definition 20 (Simplicial bisimilarity). *A binary relation $\sim \subseteq P \times P$ is a* simplicial bisimulation *if and only if for all x, y with $x \sim y$:*

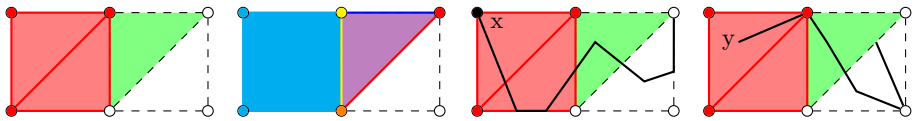
1. *for all $p \in \mathrm{AP}$, $x \in V(p) \iff y \in V(p)$;*
2. *for each* simplicial *path π_x, with $\pi_x(0) = x$, there is a simplicial path π_y with $\pi_y(0) = y$, and $\pi_x \hat{\sim} \pi_y$;*
3. *for each* simplicial *path π_y, with $\pi_y(0) = y$, there is a simplicial path π_x with $\pi_x(0) = x$, and $\pi_x \hat{\sim} \pi_y$.*

The largest simplicial bisimulation, if it exists, is called simplicial bisimilarity.

Definition 20 generalizes the classical bisimilarity of Kripke structures by considering simplicial paths instead of transitions. We can interpret this novel bisimilarity as a *spatial observational equivalence*: if a path starts from x and traverses a finite sequence of spatial regions (i.e., cells), then there is an observably equivalent path starting from y which traverses equivalent spatial regions (possibly a different number) in the same order.
Next, we state the three main facts that conclude this section.

Theorem 21. *Logical equivalence is a simplicial bisimulation.*

Theorem 22. *Each simplicial bisimulation is included in logical equivalence.*



(a) Model (b) Bisimulation (c) Path 1 (d) Path 2

Fig. 3. An example of bisimilarity. 3a) Model with atomic propositions in green and red. 3b) bisimilarity, encoded via colours (points of the same colour are bisimilar). 3c) A point x and a simplicial path starting from x. 3d) Another point y, bisimilar to x, and a simplicial path starting from y. The two paths are, in turn, bisimilar; note that these paths are also piecewise linear. The two paths cross a different set of cells, and have a different number of segments.

Corollary 23 (of Theorem 21 and 22). *In a polyhedral model, the largest simplicial bisimulation always exists, and it coincides with logical equivalence.*

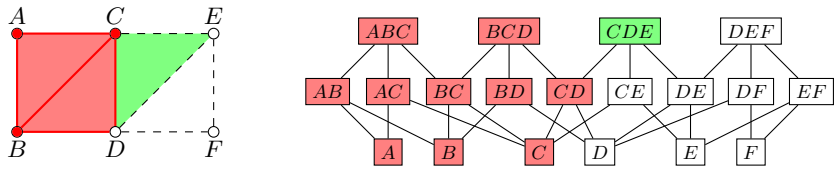
Example 24. Consider the polyhedral, 1-dimensional model with cells the points $x = -1$, $y = 0$, $z = 1$, and the open segments $s = (-1, 0)$ and $t = (0, 1)$. Consider the set of atomic propositions $\{a, b\}$. Let $V(y) = \{a\}$ and $V(z) = b$. According to topo-bisimilarity [11], which characterises the modal fragment of our language, all the points in $s \cup t$ are equivalent, as there is no modal formula telling s and t apart. However, if γ is added to the picture, let $\phi = \gamma(\neg a, b)$. The points of s do not satisfy ϕ, but the points of t do. No point of s is bisimilar to a point of t.

Example 25. In Figure 3, we propose a simple illustration of the concept of simplicial bisimilarity. Note how the two presented paths pass through a different number of equivalent cells (in a way akin to classical "stuttering" forms of bisimilarities for process calculi [7]).

5 Geometric Model Checking

Given a polyhedral model $\mathcal{X}$, this section is devoted to identifying a corresponding Kripke-style, finite model $\mathcal{M}(\mathcal{X})$. Notably, $\mathcal{M}(\mathcal{X})$ is also a topological model in the sense of [11] when equipped with the Alexandrov topology, and it is a quotient of $\mathcal{X}$ that preserves and reflects the semantics of each formula. The goal of this section is to extend the standard Kripkean semantics of modal logic to the language of SLCS, by defining a suitable semantics for γ and by showing that $\mathcal{X}$ and $\mathcal{M}(\mathcal{X})$ are logically equivalent. To do so, we introduce a suitable notion of path in $\mathcal{M}(\mathcal{X})$ corresponding to a simplicial path in $\mathcal{X}$. Model checking on $\mathcal{X}$ can then be carried on using $\mathcal{M}(\mathcal{X})$.

Definition 26. *Given a polyhedral model $\mathcal{X} = \langle P, \mathrm{K}, V\rangle$, we define the Kripke model $\mathcal{M}(\mathcal{X}) = \langle \mathrm{K}, \preceq, V'\rangle$, where $\preceq$ is the face relation of the simplicial complex* K*, and $\sigma \in V'(p)$ iff $\widetilde{\sigma} \subseteq V(p)$.*



(a) Polyhedral model $\mathcal{X}$ (b) Kripke model $\mathcal{M}(\mathcal{X})$

Fig. 4. The polyhedral model $\mathcal{X}$ of Figure 3 (4a) and its corresponding Kripke model $\mathcal{M}(\mathcal{X})$ (4b). We indicate a simplex by the set of its vertices. The accessibility relation $\preceq$ is represented via its Hasse diagram (reflexive and transitive relations are omitted). The atomic propositions g and r are indicated in green and red respectively.

We emphasize that $\mathcal{M}(\mathcal{X})$, although disregarding much of the information about a polyhedral model (e.g., the position and size of the simplexes) encodes all the information which is expressible using SLCS, while being a finite—thus computationally tractable—representation of $\mathcal{X}$. An example of polyhedral model together with its corresponding Kripke model is depicted in Figure 4.

Definition 27. *Given a polyhedral model $\mathcal{X}$, with $\mathcal{M}(\mathcal{X}) = \langle \mathrm{K}, \preceq, V' \rangle$ as in Definition 26, let $\preceq^{\pm}$ be the relation $\preceq \cup \succeq$. We say that $\pi : \{0, \dots, k\} \to S$ is a* $\pm$-path *(and we indicate it with $\pi : \{0, \dots, k\} \xrightarrow{\pm} S$) if $k \geq 2$ and $\pi(0) \preceq \pi(1) \preceq^{\pm} \pi(2) \preceq^{\pm} \dots \preceq^{\pm} \pi(k-1) \succeq \pi(k)$.*

Notice that for $\sigma, \tau \in \mathrm{K}$, the condition $\sigma \preceq^{\pm} \tau$ amounts to the cells $\widetilde{\sigma}$ and $\widetilde{\tau}$ being adjacent. So Definition 27 ensures that the path is comprised of adjacent spatial regions. Observe also that the first step of a $\pm$-path follows the relation $\preceq$, that is, either the first two simplexes in the path coincide or the dimension of the visited simplexes has to *strictly increase* along the first step; and similarly, the last step of the path is either reflexive or the dimension of the visited simplexes has to *strictly decrease*.

For typographical reasons, we will use the notation $\widetilde{\pi}(j)$ to indicate the relative interior of the simplex $\pi(j)$.

Definition 28 (Semantics on $\mathcal{M}$). *Consider $\mathcal{M}(\mathcal{X}) = \langle \mathrm{K}, \preceq, V' \rangle$. Given $\sigma \in \mathrm{K}$, satisfaction $\mathcal{M}, \sigma \vDash \phi$ over formulas ϕ is given by the following inductive clauses, where we let $[\![\phi]\!]^{\mathcal{M}}$ denote the set $\{\sigma \in \mathrm{K} \mid \mathcal{M}(\mathcal{X}), \sigma \vDash \phi\}$:*

$$
\begin{array}{ll}
\mathcal{M}(\mathcal{X}), \sigma \vDash \top & \textit{always holds} \\
\mathcal{M}(\mathcal{X}), \sigma \vDash p & \iff x \in V(p) \textit{ for } p \in \mathrm{AP} \\
\mathcal{M}(\mathcal{X}), \sigma \vDash \neg\phi & \iff \mathcal{M}(\mathcal{X}), \sigma \nvDash \phi \\
\mathcal{M}(\mathcal{X}), \sigma \vDash \phi \wedge \psi & \iff \mathcal{M}(\mathcal{X}), \sigma \vDash \phi \textit{ and } \mathcal{M}(\mathcal{X}), \sigma \vDash \psi \\
\mathcal{M}(\mathcal{X}), \sigma \vDash \Box\phi & \iff \forall \tau \in \mathrm{K}.\ \textit{if } \sigma \preceq \tau \textit{ then } \mathcal{M}(\mathcal{X}), \tau \vDash \phi \\
\mathcal{M}(\mathcal{X}), \sigma \vDash \gamma(\phi, \psi) & \iff \textit{there exists a } \pm\textit{-path } \pi : \{0, \dots, k\} \xrightarrow{\pm} \mathrm{K} \textit{ such that} \\
 & \qquad \pi(0) = \sigma,\ \pi(k) \in [\![\psi]\!]^{\mathcal{M}} \textit{ and } \pi(\{1, \dots, k-1\}) \subseteq [\![\phi]\!]^{\mathcal{M}}
\end{array}
$$

The clauses for the boolean operators and for $\Box$ are the standard interpretation of modal formulas on Kripke models (and on topological spaces, via the Alexandrov topology, see [11]). However, the semantic clause for γ involves a rather complex notion of path: this is a clause tailored to simulate the behaviour of γ on polyhedral models. An example of a $\pm$-path corresponding to a topological path is shown in Figure 5. The following theorem shows that this is indeed the correct notion to consider.

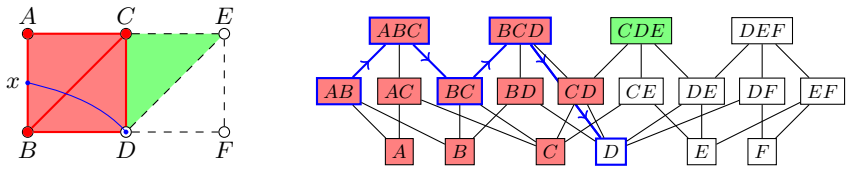


Fig. 5. On the left, a simplicial path π (in blue) witnessing $x \in [\![\gamma(r, \neg(r \vee g))]\!]^{\mathcal{X}}$; on the right, the corresponding $\pm$-path π' (again in blue) witnessing $AB \in [\![\gamma(r, \neg(r \vee g))]\!]^{\mathcal{M}}$. Only $\pi(0)$ belongs to the simplex AB and the path enters immediately the simplex ABC: this is possible since $AB \preceq ABC$. Likewise, the path ends with a transition from the simplex BCD to the simplex D, which is possible since $BCD \succeq D$. These are exactly the requirements on the first and last steps of a $\pm$-path.

Theorem 29. *Let x be a point of P. Let $\sigma \in \mathrm{K}$ be the unique simplex such that $x \in \widetilde{\sigma}$. For every formula ϕ of SLCS we have $\mathcal{X}, x \vDash \phi \iff \mathcal{M}(\mathcal{X}), \sigma \vDash \phi$.*

6 PolyLogicA: a Model Checker for Polyhedra

Based on the theory of Section 5, we developed the prototype model checker PolyLogicA: a *Polyhedral Logic-based Analysis tool*. The tool is written in the functional language FSharp[8]. PolyLogicA is Free and Open Source Software, distributed under the Apache 2.0 license[9].

In this section, we provide a functional description of the tool. For space reasons, implementation details, including the pseudo-code for model checking the reachability operator, and a discussion of the complexity of the encoding from simplicial models to Kripke structures, and of the (global) model checking algorithm, are given in Appendix B. Here we only mention that, once the dimension d of the space is fixed, as in our intended applications to processing of 3D meshes, in our current implementation the time complexity of model checking a formula f—including the intermediate computation of the Kripke structure starting from a standard 3D mesh format—is in $\mathcal{O}(n \cdot h)$ where n is the number of simplexes and h the number of subformulas of f. Such complexity grows

[8] See `https://www.fsharp.org`

[9] The tool is currently available in a branch of the main VoxLogicA repository, see `https://github.com/vincenzoml/VoxLogicA`.

exponentially in d. The design space for algorithms that scale better with d, possibly exploiting specialised data structures (see e.g. the recent work [14]) will be explored in future research, depending on the considered use case.

A PolyLogicA specification consists of a text file that can make use of four commands: `let`, for declaring functions and constants; `import`, for importing libraries of such declarations; `load`, to specify the file to be loaded as a model, `save`, to specify the logic formulas that need to be computed, and saved, possibly making use of previous `let` declarations.

Models are required to be based on a fixed simplicial complex. A model file uses a custom `json`-based[10] format. The information contained in the file consists of: a list p of d-dimensional vectors, denoting the coordinates of the 0-cells of the polyhedron, and implicitly specifying the dimension of the underlying space $\mathbb{R}^d$; a list of atomic proposition identifiers; a list of simplexes. Each simplex is specified by the list of the indexes of its vertices in p, and its specification also contains the list of atomic propositions holding at the simplex.

Logic formulas are just a concrete syntax for SLCS. Currently it does not implement additional extra-logical operators (contrary to VoxLogicA, which also implements imaging primitives). PolyLogicA is in spirit a *global, explicit-state* model checker, that is, the set of simplexes satisfying a given formula is computed and returned at once. The tool outputs a list in `json` format, having an element for each formula ϕ that the specification requires to be checked. Each element of such list contains in turn a list representing the truth values of ϕ at each point (simplex) of the input model. Finally, a simple 3D, web-based visualizer has been implemented along the prototype (see the screenshot in Figure 6), which will be refined in future work.

So far, PolyLogicA has been experimented on some example 3D medical images, such as the one in Figure 6(b). The mesh visualized is a simplified version (circa 20000 polygons) of a larger mesh (circa 400000 polygons). A custom converter has been implemented to obtain a `json` file from the input `obj` mesh.[11] On a desktop machine equipped with an Intel core i7 7700 cpu and 16gb of RAM, the execution time to compute the properties shown in the figure is of circa 4.6 seconds, of which 3.5 seconds are spent parsing the (very large) `json` file, circa 0.7 seconds are spent computing the Kripke model, and the actual model checking procedure takes just 0.4 seconds. The tool also manages to compute the same specification on the original image, taking total time of 128 seconds, of which 56 seconds are spent in parsing, 65 in computing the Kripke model, and model checking takes 7 seconds (indeed, memory management for very large files can certainly make the execution time scale less-than-ideally with the problem size, especially since garbage collection can be triggered several times along the computations). Future work will also include implementing a fast loader for `obj` meshes, in order to eliminate the parsing of (very large) `json` files, and optimizing the translation from simplicial complexes to Kripke models, which currently

[10] See `https://www.json.org/`.

[11] *Wavefront* `obj` is a widely used file format for 3D meshes. See `https://en.wikipedia.org/wiki/Wavefront_.obj_file`

exploits purely-functional data structures for ease of prototyping. We note in passing that the intermediate Kripke model could be cached for speeding up the execution of several analyses.

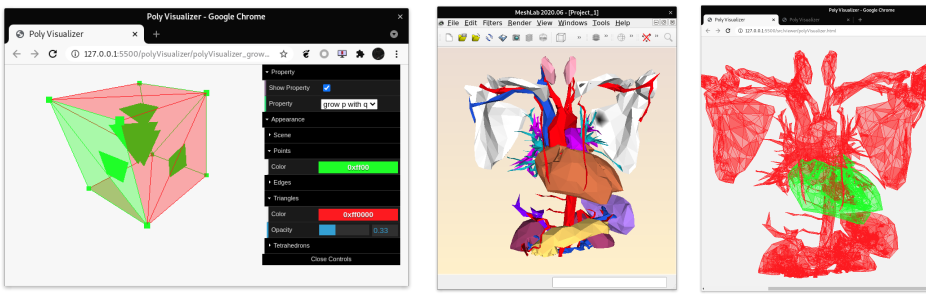
The current version of the tool has been implemented sharing part of the code base with VoxLogicA. PolyLogicA inherits from its parent tool the multi-threaded, memoizing computation engine, and the parser for the input language of the tool. Basically, after expanding `let` bindings, each formula is converted into a directed acyclic graph where nodes are *tasks*, and arcs are *dependencies*. Each task is a basic logical primitive, to be applied to specific arguments. Task A depends upon task B if and only if the result of B is an argument of A. The implementation guarantees that, while being constructed, the task graph is kept *minimal* in the sense that the same primitive on the same arguments will never be computed twice. After having been constructed, the task graph is executed in parallel as much as possible, exploiting the available CPU cores.

7 Conclusions and Future Work

In this work, we presented a novel methodology to verify spatial properties of polyhedral spaces, stemming from topological modal logics enhanced with reachability. We showed that a natural notion of bisimilarity characterises logical equivalence, and we demonstrated a model checking algorithm for the reachability operator based on a particular finite representation. For the purpose, we developed a prototype model checker which is already able to efficiently compute the interpretation of formulas on 3D meshes. Future work will span several theoretical and applied directions.

Regarding bisimilarity, by looking at the *simplicial* paths that are used in Definition 20, one may wonder what logical operators (alternative to γ), and in what classes of models, can be characterised by lifting the restrictions or completely changing the kind of paths that are used therein. Furthermore, bisimilarity hints at *minimization* in order to reduce the complexity of the analysis. The preliminary results presented in [26], including the tool *MiniLogicA* could be useful in this research direction. Note that the quotient mapping each simplex in P to a point in K in Definition 28 is *open*, thus it preserves and reflects logical equivalence of the modal fragment of our language; additionally, it preserves and reflects logical equivalence of the full language, thus simplicial bisimilarity. Not all open maps do so (just consider, e.g. the quotient with respect to classical modal logical equivalence). In future work, we plan to formalise the conditions on an arbitrary open map that make it preserve and reflect simplicial bisimilarity. The relationship between spatial logics and temporal logics, and related bisimilarities [37] is also of interest, and in particular, comparing path-based spatial notions such as simplicial bisimilarity, to the so-called *stuttering* equivalences, and their associated minimization algorithms (see e.g. [16,32]).

Spatio-temporal model checking in the style of [31,19] is a planned future development, the simplest case being the one where the underlying polyhedron does not change over time, and only the valuation of atomic propositions depends



(a) Visualizer (b) Simple 3D medical image (c) Heart

(d) Liver (e) Some veins (f) Selected vein

Fig. 6. PolyLogicA and its visualizer. 6a) Screenshot of the prototype visualizing the result of an analysis. 6b) A 3D medical illustration, courtesy of `www.sketchfab.com` (copyright: COEUR et vaissaaux by Chair_Digital_Anatomy – The Unesco Chair of digital anatomy (Paris University) – is licensed under Creative Commons Attribution, see `https://creativecommons.org/licenses/by/4.0/legalcode`), visualized using MeshLab [27]. PolyLogicA is used to segment the heart (6c), liver (6d), and some veins (6e), using face colours, and then to segment a specific vein (the one that reaches the liver) using a reachability predicate (6f). Such regions are shown in green.

upon the temporal state of a system. More complex forms of dynamic spatial structures where the underlying polyhedron evolves over time are also of interest.

A promising application of PolyLogicA is fully automated, declarative analysis of 3D *meshes*. Clearly, we foresee 3D medical imaging to be a promising landscape for future research. Furthermore, note that 3D meshes play a central role in several fields, including e.g. architecture and computer-aided design (CAD), or the entertainment industry (consider 3D games or 3D animation movies).

Implementation-wise, *GPU computing* could provide a computational boost to PolyLogicA. See [17] for a GPU implementation of the parent tool VoxLogicA. Finally, a user interface could be useful to explore large datasets, and to better visualize the interpretation of logic formulas, possibly exploiting results in [15] for validation.

References

1. Adam-Day, S.: Polyhedral completeness in intermediate and modal logics (2019), master's Thesis, Available as ILLC report: MoL-2019-08
2. Adam-Day, S., Bezhanishvili, N., Gabelaia, D., Marra, V.: The nerve criterion and polyhedral completeness of intermediate logics (2020), submitted. Available as ILLC prepirint: PP-2020-22
3. Aiello, M., Pratt-Hartmann, I., Benthem, van, J.: Handbook of Spatial Logics. Springer (2007). https://doi.org/10.1007/978-1-4020-5587-4
4. Aiello, M., Pratt-Hartmann, I., van Benthem, J.: What is spatial logic? In: Handbook of Spatial Logics [3], pp. 1–11. https://doi.org/10.1007/978-1-4020-5587-4_1
5. Alur, R.: Formal verification of hybrid systems. In: Proceedings of the 11th International Conference on Embedded Software, EMSOFT 2011, part of the Seventh Embedded Systems Week, ESWeek 2011, Taipei, Taiwan, October 9-14, 2011. pp. 273–278. ACM (2011). https://doi.org/10.1145/2038642.2038685
6. Alur, R., Giacobbe, M., Henzinger, T.A., Larsen, K.G., Mikucionis, M.: Continuous-time models for system design and analysis. In: Computing and Software Science - State of the Art and Perspectives, Lecture Notes in Computer Science, vol. 10000, pp. 452–477. Springer (2019). https://doi.org/10.1007/978-3-319-91908-9_22
7. Baier, C., Katoen, J.: Principles of model checking. MIT Press (2008)
8. Belmonte, G., Broccia, G., Vincenzo, C., Latella, D., Massink, M.: Feasibility of spatial model checking for nevus segmentation. In: FORMALISE: International Conference on Formal Methods in Software Engineering. p. To appear (2021)
9. Belmonte, G., Ciancia, V., Latella, D., Massink, M.: Innovating medical image analysis via spatial logics. In: From Software Engineering to Formal Methods and Tools, and Back - Essays Dedicated to Stefania Gnesi on the Occasion of Her 65th Birthday. Lecture Notes in Computer Science, vol. 11865, pp. 85–109. Springer (2019). https://doi.org/10.1007/978-3-030-30985-5_7
10. 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. LNCS, vol. 11427, pp. 281–298. Springer (2019). https://doi.org/10.1007/978-3-030-17462-0_16
11. Benthem, van, J., Bezhanishvili, G.: Modal logics of space. In: Handbook of Spatial Logics [3], pp. 217–298. https://doi.org/10.1007/978-1-4020-5587-4_5
12. Bezhanishvili, N., Marra, V., McNeill, D., Pedrini, A.: Tarski's theorem on intuitionistic logic, for polyhedra. Ann. Pure Appl. Log. **169**(5), 373–391 (2018)
13. Bogomolov, S., Frehse, G., Giacobbe, M., Henzinger, T.A.: Counterexample-guided refinement of template polyhedra. In: Tools and Algorithms for the Construction and Analysis of Systems - 23rd International Conference, TACAS 2017. Lecture Notes in Computer Science, vol. 10205, pp. 589–606 (2017). https://doi.org/10.1007/978-3-662-54577-5_34
14. Boissonnat, J.D., Maria, C.: The simplex tree: An efficient data structure for general simplicial complexes. Algorithmica **70**(3), 406–427 (Nov 2014). https://doi.org/10.1007/s00453-014-9887-3
15. Broccia, G., Milazzo, P., Ölveczky, P.C.: Formal modeling and analysis of safety-critical human multitasking. Innov. Syst. Softw. Eng. **15**(3-4), 169–190 (2019). https://doi.org/10.1007/s11334-019-00333-7

16. Browne, M.C., Clarke, E.M., Grumberg, O.: Characterizing finite kripke structures in propositional temporal logic. Theor. Comput. Sci. **59**, 115–131 (1988). https://doi.org/10.1016/0304-3975(88)90098-9, https://doi.org/10.1016/0304-3975(88)90098-9
17. Bussi, L., Ciancia, V., Gadducci, F.: A spatial model checker in GPU. In: FORTE 2021 - 41st International Conference on Formal Techniques for Distributed Objects, Components, and Systems. Lecture Notes in Computer Science, vol. To appear. Springer (2021)
18. Ciancia, V., Gilmore, S., Latella, D., Loreti, M., Massink, M.: Data verification for collective adaptive systems: Spatial model-checking of vehicle location data. In: Eighth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW. pp. 32–37. IEEE Computer Society (2014). https://doi.org/10.1109/SASOW.2014.16
19. Ciancia, V., Grilletti, G., Latella, D., Loreti, M., Massink, M.: An experimental spatio-temporal model checker. In: Software Engineering and Formal Methods - SEFM 2015 Collocated Workshops. Lecture Notes in Computer Science, vol. 9509, pp. 297–311. Springer (2015). https://doi.org/10.1007/978-3-662-49224-6_24
20. 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). https://doi.org/10.1007/978-3-662-44602-7_18
21. 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). https://doi.org/10.2168/LMCS-12(4:2)2016, http://lmcs.episciences.org/2067
22. Ciancia, V., Latella, D., Massink, M., Paškauskas, R., Vandin, A.: A tool-chain for statistical spatio-temporal model checking of bike sharing systems. In: Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 7th International Symposium, ISoLA 2016, Part I. Lecture Notes in Computer Science, vol. 9952, pp. 657–673 (2016). https://doi.org/10.1007/978-3-319-47166-2_46
23. 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
24. Ciancia, V., Latella, D., Loreti, M., Massink, M.: Spatial logic and spatial model checking for closure spaces. In: 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. Lecture Notes in Computer Science, vol. 9700, pp. 156–201. Springer (2016). https://doi.org/10.1007/978-3-319-34096-8_6
25. Ciancia, V., Latella, D., Massink, M., Paškauskas, 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
26. Ciancia, V., Latella, D., Massink, M., de Vink, E.P.: Towards spatial bisimilarity for closure models: Logical and coalgebraic characterisations. CoRR **abs/2005.05578** (2020), https://arxiv.org/abs/2005.05578

27. Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G.: MeshLab: an Open-Source Mesh Processing Tool. In: Scarano, V., Chiara, R.D., Erra, U. (eds.) Eurographics Italian Chapter Conference. The Eurographics Association (2008). https://doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136
28. Galton, A.: The mereotopology of discrete space. In: Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science, Lecture Notes in Computer Science, vol. 1661, pp. 251–266. Springer (1999). https://doi.org/10.1007/3-540-48384-5_17, http://dx.doi.org/10.1007/3-540-48384-5_17
29. Galton, A.: Discrete mereotopology. In: Calosi, C., Graziani, P. (eds.) Mereology and the Sciences: Parts and Wholes in the Contemporary Scientific Context, pp. 293–321. Springer International Publishing (2014). https://doi.org/10.1007/978-3-319-05356-1_11, https://doi.org/10.1007/978-3-319-05356-1_11
30. Girard, A., Guernic, C.L.: Zonotope/hyperplane intersection for hybrid systems reachability analysis. In: Hybrid Systems: Computation and Control, 11th International Workshop, 2008. Lecture Notes in Computer Science, vol. 4981, pp. 215–228. Springer (2008). https://doi.org/10.1007/978-3-540-78929-1_16
31. Grilletti, G.: Spatio-Temporal Model Checking: Explicit and Abstraction-Based Methods. Master's thesis, University of Pisa (2016)
32. Groote, J.F., Jansen, D.N., Keiren, J.J.A., Wijs, A.: An $O(m\log n)$ algorithm for computing stuttering equivalence and branching bisimulation. ACM Trans. Comput. Log. **18**(2), 13:1–13:34 (2017). https://doi.org/10.1145/3060140, https://doi.org/10.1145/3060140
33. Grosu, R., Smolka, S., Corradini, F., Wasilewska, A., Entcheva, E., Bartocci, E.: Learning and detecting emergent behavior in networks of cardiac myocytes. Commun. ACM **52**(3), 97–105 (2009)
34. Haghighi, I., Jones, A., Kong, Z., Bartocci, E., Grosu, R., Belta, C.: Spatel: A novel spatial-temporal logic and its applications to networked systems. In: Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control. pp. 189–198. HSCC '15, ACM (2015). https://doi.org/10.1145/2728606.2728633
35. Henzinger, T.A.: The theory of hybrid automata. In: Verification of Digital and Hybrid Systems, pp. 265–292. Springer (2000). https://doi.org/10.1007/978-3-642-59615-5_13
36. Henzinger, T.A., Ho, P.: HYTECH: the cornell hybrid technology tool. In: Hybrid Systems II, Proceedings of the Third International Workshop on Hybrid Systems. Lecture Notes in Computer Science, vol. 999, pp. 265–293. Springer (1994). https://doi.org/10.1007/3-540-60472-3_14
37. Kurtonina, N., de Rijke, M.: Bisimulations for temporal logic. J. Log. Lang. Inf. **6**(4), 403–425 (1997). https://doi.org/10.1023/A:1008223921944, https://doi.org/10.1023/A:1008223921944
38. Levine, J.A., Paulsen, R.R., Zhang, Y.: Mesh processing in medical-image analysis?a tutorial. IEEE Computer Graphics and Applications **32**(5), 22–28 (2012). https://doi.org/10.1109/MCG.2012.91
39. Linker, S., Papacchini, F., Sevegnani, M.: Analysing spatial properties on neighbourhood spaces. In: Esparza, J., Král', D. (eds.) 45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic. LIPIcs,

vol. 170, pp. 66:1–66:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). https://doi.org/10.4230/LIPIcs.MFCS.2020.66, https://doi.org/10.4230/LIPIcs.MFCS.2020.66
40. Maunder, C.R.F.: Algebraic topology. Cambridge University Press. (1980)
41. Nenzi, L., Bortolussi, L., Ciancia, V., Loreti, M., Massink, M.: Qualitative and Quantitative Monitoring of Spatio-Temporal Properties with SSTL. Logical Methods in Computer Science **14**(4), 1–38 (2018), DOI 10.23638/LMCS-14(4:2)2018. Published on line: 23 Oct. 2018. ISSN: 1860-5974
42. Sankaranarayanan, S., Dang, T., Ivancic, F.: Symbolic model checking of hybrid systems using template polyhedra. In: Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008. Lecture Notes in Computer Science, vol. 4963, pp. 188–202. Springer (2008). https://doi.org/10.1007/978-3-540-78800-3_14
43. Tsigkanos, C., Kehrer, T., Ghezzi, C.: Modeling and verification of evolving cyber-physical spaces. In: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2017. pp. 38–48. ACM (2017). https://doi.org/10.1145/3106237.3106299, https://doi.org/10.1145/3106237.3106299
44. Van Benthem, J.: Correspondence Theory, pp. 167–247. Springer Netherlands, Dordrecht (1984). https://doi.org/10.1007/978-94-009-6259-0_4

A Additional lemmas, and Proofs

Proposition A.1. *Consider the operator ρ of [10]. We have $\mathcal{X}, x \vDash \rho\,\psi[\phi] \iff \mathcal{X}, x \models \psi \vee \gamma(\phi, \psi)$, and $\mathcal{X}, x \vDash \gamma(\phi, \psi) \iff \mathcal{X}, x \vDash \rho\,(\phi \wedge \rho\,\psi[\phi])[\phi]$.*

Proof (Proposition A.1). We note in passing that the following proof (and the definition of ρ and γ) generalise to arbitrary topological models. We first recall the formal definition of ρ, which uses right-open paths (that is, total continuous functions having $\mathbb{R}_{\geq 0}$ as a domain). Note that in [10], ρ was defined only on discrete spaces. However, following the direction of [21], the definition applies in a natural way to continuous spaces as follows: $\mathcal{X}, x \vDash \rho\,\psi\,[\phi]$ whenever there is a right-open path $\pi : \mathbb{R}_{\geq 0} \to P$ and an index r such that $\pi(0) = x$, $\pi(r) \in [\![\psi]\!]^{\mathcal{X}}$ and $\pi((0, r)) \subseteq [\![\phi]\!]^{\mathcal{X}}$. We proceed by proving the four implications in the statement separately.

First, we show that $\mathcal{X}, x \models \rho\,\psi[\phi]$ implies $\mathcal{X}, x \models \psi \vee \gamma(\phi, \psi)$. If $\mathcal{X}, x \models \rho\,\psi[\phi]$ then there is right-open path π and $\ell \in \mathbb{R}_{\geq 0}$ such that $\pi(0) = x, \pi((0, \ell)) \subseteq [\![\phi]\!]^{\mathcal{X}}$ and $\pi(\ell) \in [\![\psi]\!]^{\mathcal{X}}$. We consider two distinct cases:
Case 1: $\ell = 0$. In this case $x \in [\![\psi]\!]^{\mathcal{X}}$ and so $\mathcal{X}, x \models \psi \vee \gamma(\phi, \psi)$.
Case 2: $\ell \neq 0$. In this case, let π' with $\pi'(r) = \pi(r\ell)$ for all $r \in [0, 1]$. We have $\pi'(0) = \pi(0) = x$, $\pi'((0, 1)) = \pi((0, \ell)) \subseteq [\![\phi]\!]^{\mathcal{X}}$ and $\pi'(1) = \pi(\ell) \in [\![\psi]\!]^{\mathcal{X}}$. This means that $\mathcal{X}, x \models \gamma(\phi, \psi)$, and so $\mathcal{X}, x \models \psi \vee \gamma(\phi, \psi)$.

We now show that $\mathcal{X}, x \models \psi \vee \gamma(\phi, \psi)$ implies $\mathcal{X}, x \models \rho\,\psi[\phi]$. Suppose that $\mathcal{X}, x \models \psi \vee \gamma(\phi, \psi)$. We consider two distinct cases:
Case 1: $\mathcal{X}, x \models \psi$. In this case it trivially holds $\mathcal{X}, x \models \rho\,\psi[\phi]$ (for any right-open path starting at x, just consider $\ell = 0$).

Case 2: $\mathcal{X}, x \models \gamma(\phi, \psi)$. In this case there is a path π such that $\pi(0) = x$, $\pi((0,1)) \subseteq [\![\phi]\!]^{\mathcal{X}}$ and $\pi(1) \in [\![\psi]\!]^{\mathcal{X}}$. We obtain $\mathcal{X}, x \models \rho\,\psi[\phi]$ by taking $\ell = 1$ in the definition of the semantic clause of ρ.

We now show that $\mathcal{X}, x \models \gamma(\phi, \psi)$ implies $\mathcal{X}, x \models \rho\,(\phi \wedge \rho\,\psi[\phi])[\phi]$. Suppose $\mathcal{X}, x \models \gamma(\phi, \psi)$, that is, there is a path π such that $\pi(0) = x$, $\pi((0,1)) \subseteq [\![\phi]\!]^{\mathcal{X}}$ and $\pi(1) \in [\![\psi]\!]^{\mathcal{X}}$. Since π is a total continuous function, there is an $\epsilon \in (0,1)$ such that $\pi(\epsilon) \in [\![\phi]\!]^{\mathcal{X}}$. We define two additional paths $\pi_1(t) = \pi(t\epsilon)$ and $\pi_2(t) = \pi((1-\epsilon)t + \epsilon)$. Notice that $\pi_2(0) = \pi(\epsilon) \in [\![\phi]\!]^{\mathcal{X}}$, $\pi_2((0,1)) = \pi((\epsilon,1)) \subseteq [\![\phi]\!]^{\mathcal{X}}$ and $\pi_2(1) = \pi(1) \in [\![\psi]\!]^{\mathcal{X}}$: this shows that $\mathcal{X}, \pi(\epsilon) \models \phi \wedge \rho\,\psi[\phi]$. Moreover, we have $\pi_1(0) = \pi(0) = x, \pi_1((0,1)) = \pi((0,\epsilon)) \subseteq [\![\phi]\!]^{\mathcal{X}}$ and $\pi_1(1) = \pi_2(0) \in [\![\phi \wedge \rho\,\psi[\phi]]\!]^{\mathcal{X}}$: this shows that $\mathcal{X}, x \models \rho\,(\phi \wedge \rho\,\psi[\phi])[\phi]$.

Finally, we show that $\mathcal{X}, x \models \rho\,(\phi \wedge \rho\,\psi[\phi])[\phi]$ implies $\mathcal{X}, x \models \gamma(\phi, \psi)$. Suppose that $\mathcal{X}, x \models \rho\,(\phi \wedge \rho\,\psi[\phi])[\phi]$, that is, there is a right-open path π_1 and a value $\ell_1 \in \mathbb{R}_{\geq 0}$ such that $\pi_1(0) = x, \pi_1((0,\ell_1)) \subseteq [\![\phi]\!]^{\mathcal{X}}$ and $\pi_1(\ell_1) \in [\![\phi \wedge \rho\,\psi[\phi]]\!]^{\mathcal{X}} = [\![\phi]\!]^{\mathcal{X}} \cap [\![\rho\,\psi[\phi]]\!]^{\mathcal{X}}$. By the last condition, there is a right-open path π_2 and a value $\ell_2 \in \mathbb{R}_{\geq 0}$ such that $\pi_2(0) = \pi_1(\ell_1)$, $\pi_2((0,\ell_2)) \subseteq [\![\phi]\!]^{\mathcal{X}}$ and $\pi_2(\ell_2) \in [\![\psi]\!]^{\mathcal{X}}$. We consider two distinct cases:
Case 1: $l_1 = l_2 = 0$. In this case $\pi_1(l_1) = \pi_2(l_2) = x$, and so $x \in [\![\phi]\!]^{\mathcal{X}} \cap [\![\psi]\!]^{\mathcal{X}}$. In this case $\mathcal{X}, x \vDash \gamma(\phi, \psi)$ holds trivially.
Case 2: $l_1 > 0$ or $l_2 > 0$. In this case define the path π' by imposing $\pi'(t) = \pi_1(2tl_1)$ and $\pi'(\frac{1}{2} + t) = \pi_2(2tl_2)$ for $t \in [0, \frac{1}{2}]$ (notice that the path is well-defined since $\pi'(\frac{1}{2}) = \pi_1(l_1) = \pi_2(0)$). Clearly $\pi'(0) = x$. Moreover, $\pi'((0,1)) = \pi'((0,\frac{1}{2}]) \cup \pi'((\frac{1}{2},1)) = \pi_1((0,l_1)) \cup \pi_2((0,l_2)) \subseteq [\![\phi]\!]^{\mathcal{X}}$ (under the assumption that $l_1 > 0$ or $l_2 > 0$). And finally $\pi'(1) = \pi_2(l_2) \in [\![\psi]\!]^{\mathcal{X}}$. This shows that $\mathcal{X}, x \vDash \gamma(\phi, \psi)$, as desired.

Proposition A.2. *Let σ be a non-empty simplex and consider $x \in \widetilde{\sigma}$ and $y \in \sigma$. Then there is a* linear *path (a segment in $\mathbb{R}^d$) $\pi : [0,1] \to \sigma$ with $\pi(0) = x$, $\pi(1) = y$ and $\pi([0,1)) \subseteq \widetilde{\sigma}$; more precisely, $\pi(t) = ty + (1-t)x$.*

Proof (Proposition A.2). Let $V = \{v_0, \ldots, v_n\}$ be the set of vertices of σ. By definition of σ, x and y are in the convex hull of V, that is, there exist λ_i and δ_i such that $x = \sum_{i=0}^{n} \lambda_i v_i$ and $y = \sum_{i=0}^{n} \delta_i v_i$. Since $x \in \widetilde{\sigma}$, every λ_i is strictly greater than 0. For π as defined in the statement we have $\pi(t) = \sum_{i=0}^{n} (t\delta_i + (1-t)\lambda_i)\, v_i$. This function is clearly continuous (thus a path), and we have $\pi(0) = x$ and $\pi(1) = y$. Moreover, for every $t \in [0,1)$ and every $i \leq n$ we have $t\delta_i + (1-t)\lambda_i \geq (1-t)\lambda_i > 0$, and so $\pi([0,1)) \subseteq \widetilde{\sigma}$.

Lemma A.3. *Let* K *be a simplicial complex and $\sigma, \tau \in$ K. Then $\sigma \preceq \tau$ iff $\widetilde{\sigma} \subseteq \mathcal{C}(\widetilde{\tau})$ iff $\widetilde{\sigma} \cap \mathcal{C}(\widetilde{\tau}) \neq \emptyset$.*

Proof (Lemma A.3). First, we show that $\sigma \preceq \tau$ implies $\widetilde{\sigma} \subseteq \mathcal{C}(\widetilde{\tau})$. In fact, if $\sigma \preceq \tau$ we have $\sigma \subseteq \tau$, and so $\widetilde{\sigma} \subseteq \sigma \subseteq \tau = \mathcal{C}(\widetilde{\tau})$. Secondly, note that since cells are nonempty sets we have that $\widetilde{\sigma} \subseteq \mathcal{C}(\widetilde{\tau})$ implies $\widetilde{\sigma} \cap \mathcal{C}(\widetilde{\tau}) \neq \emptyset$. Finally, we show that $\widetilde{\sigma} \cap \mathcal{C}(\widetilde{\tau}) \neq \emptyset$ implies $\sigma \preceq \tau$, concluding the proof. Notice that

$\mathcal{C}(\widetilde{\tau}) = \tau = \bigcup\{\widetilde{\tau}' \mid \tau' \preceq \tau\}$. So by Lemma 4 we have that either $\widetilde{\sigma} \in \{\widetilde{\tau}' \mid \tau' \preceq \tau\}$ or $\widetilde{\sigma} \cap \mathcal{C}(\widetilde{\tau}) = \emptyset$. Since the latter is not the case by assumption, we conclude $\sigma \preceq \tau$.

Lemma A.4. *Let π_1 and π_2 be two paths, with $\pi_1(1) = \pi_2(0)$; let $x, y \in [0,1]$, with $x < y$. Define in the obvious way the* concatenation *$\pi_1;\pi_2$, with $\pi_1;\pi_2(0) = \pi_1(0)$ and $\pi_1;\pi_2(1) = \pi_2(1)$ and the sub-path ${\pi_1}^{[x,y]}$, with ${\pi_1}^{[x,y]}(0) = \pi_1(x)$ and ${\pi_1}^{[x,y]}(1) = \pi_1(y)$. We have that: whenever π_1 and π_2 are piecewise-linear (simplicial), also $\pi_1;\pi_2$ is piecewise linear (simplicial); whenever π_1 is piecewise linear (simplicial), also ${\pi_1}^{[x,y]}$ is piecewise linear (simplicial). Furthermore, for any two (additional) paths π_1', π_2' with $\pi_1'(1) = \pi_2'(0)$, and relation R, if $\pi_1 \hat{R} \pi_1'$ and $\pi_2 \hat{R} \pi_2'$, then $\pi_1;\pi_2 \hat{R} \pi_1';\pi_2'$.*

Proof (Lemma A.4). We omit the proof, which is straightforward.

Proof (Lemma 9). Just note that K does not appear in Definition 8.

Proof (Lemma 11). We prove the result by induction on the structure of ϕ. Since most of the cases follow easily from the semantic clauses of the logic, we show only the cases for $\phi = \Box\psi$ and $\phi = \gamma(\psi, \chi)$.

Case $\phi = \Box\psi$: By inductive hypothesis, $[\![\psi]\!]^{\mathcal{X}}$ is a union of cells of $\widetilde{\mathrm{K}}$, and by Lemma 4 also $P \setminus [\![\psi]\!]^{\mathcal{X}}$ is a union of cells. Since there are only finitely many cells in $\widetilde{\mathrm{K}}$, we have $\mathcal{C}(P \setminus [\![\psi]\!]^{\mathcal{X}}) = \bigcup\{\mathcal{C}(\widetilde{\sigma}) \mid \widetilde{\sigma} \subseteq P \setminus [\![\psi]\!]^{\mathcal{X}}\}$. And since $\mathcal{C}(\widetilde{\sigma}) = \sigma = \bigcup\{\widetilde{\tau} \mid \tau \preceq \sigma\}$, it follows that $\mathcal{C}(P \setminus [\![\psi]\!]^{\mathcal{X}}) = \bigcup\{\widetilde{\tau} \mid \exists \sigma.\, \tau \preceq \sigma \text{ and } \widetilde{\sigma} \subseteq P \setminus [\![\psi]\!]^{\mathcal{X}}\}$, that is, $\mathcal{C}(P \setminus [\![\psi]\!]^{\mathcal{X}})$ is a union of cells. To conclude, by Lemma 4 we have that $[\![\phi]\!]^{\mathcal{X}} = \mathcal{I}([\![\psi]\!]^{\mathcal{X}}) = P \setminus \mathcal{C}(P \setminus [\![\psi]\!]^{\mathcal{X}})$ is again a union of cells.

Case $\phi = \gamma(\psi, \chi)$: Suppose that $\mathcal{X}, x \vDash \gamma(\phi, \psi)$: we aim to show that $\mathcal{X}, y \vDash \gamma(\phi, \psi)$. This means that there exists a path π such that $\pi(0) = x$, $\pi((0,1)) \subseteq [\![\phi]\!]^{\mathcal{X}}$ and $\pi(1) \in [\![\psi]\!]^{\mathcal{X}}$. By inductive hypothesis $[\![\phi]\!]^{\mathcal{X}}$ is a union of cells, and since $[\![\phi]\!]^{\mathcal{X}} \cap \pi((0,1))$ is not empty also the set $\mathit{\Pi} = \{\widetilde{\tau} | \exists r \in (0,1).\pi(r) \in \widetilde{\tau} \subseteq [\![\phi]\!]^{\mathcal{X}}\}$ is not empty.

Since $\pi((0,1)) \subseteq \bigcup \mathit{\Pi}$, we have $x \in \mathcal{C}(\pi((0,1))) \subseteq \bigcup\{\tau | \exists r \in (0,1).\pi(r) \in \widetilde{\tau} \subseteq [\![\phi]\!]^{\mathcal{X}}\}$. Thus there exists a value $r \in (0,1)$ and a cell $\widetilde{\tau}$ such that $x \in \tau$ and $\pi(r) \in \widetilde{\tau}$. Since $x \in \widetilde{\sigma} \cap \mathcal{C}(\widetilde{\tau})$, by Lemma A.3 we have $\widetilde{\sigma} \subseteq \mathcal{C}(\widetilde{\tau}) = \tau$ and consequently $y \in \tau$. By Proposition A.2 (modulo inverting and reparametrising the path) there exists a path $\pi' : [0, r] \to \tau$ such that $\pi'(0) = y$, $\pi'(r) = \pi(r)$ and $\pi'((0,r)) \subseteq \widetilde{\tau} \subseteq [\![\phi]\!]^{\mathcal{X}}$. If we extend π' by imposing $\pi'(t) = \pi(t)$ for $t \in (r, 1]$, we obtain a path such that $\pi'(0) = y$, $\pi'((0,1)) \subseteq [\![\phi]\!]^{\mathcal{X}}$ and $\pi'(1) \in [\![\psi]\!]^{\mathcal{X}}$. In particular π' witnesses that $\mathcal{X}, y \vDash \gamma(\phi, \psi)$, as desired.

Proof (Proposition 12). Fix K coherent with $\mathcal{X}$. By Lemma 11, each equivalence class is a union of distinct cells of $\widetilde{\mathrm{K}}$. But since there are only finitely many cells, there are finitely many equivalence classes too. For two distinct equivalence classes C and C', let $\phi^{C,C'}$ be a formula satisfied by the elements of C but not by the elements in C'—such a formula exists by definition of $\equiv$. Then the formula $\phi^{C} = \bigwedge_{C' \neq C} \phi^{C,C'}$ is satisfied only by elements in C, as desired.

Proof (Theorem 13). Recall the definition of $\Diamond\phi := \neg\Box\neg\phi$ and that $\mathcal{X}, x \models \Diamond\phi \iff x \in \mathcal{C}(\llbracket\phi\rrbracket^{\mathcal{X}})$ (see e.g. [11]), where $\mathcal{C}$ is the topological closure operator. Thus, we prove the equivalent statement $\mathcal{X}, x \vDash \Diamond\phi \iff \mathcal{X}, x \vDash \gamma(\phi, \top)$.

If $\mathcal{X}, x \models \gamma(\phi, \top)$, by definition, there is a path π with $\pi(0) = x$ and $\pi((0,1)) \subseteq \llbracket\phi\rrbracket^{\mathcal{X}}$. In particular we have $x = \pi(0) \in \mathcal{C}(\,\pi((0,1))\,)$ by properties of paths, and $\mathcal{C}(\,\pi((0,1))\,) \subseteq \mathcal{C}(\llbracket\phi\rrbracket^{\mathcal{X}})$ by monotonicity of $\mathcal{C}$. Therefore $x \in \mathcal{C}(\llbracket\phi\rrbracket^{\mathcal{X}})$, which amounts to $\mathcal{X}, x \vDash \Diamond\phi$.

If $\mathcal{X}, x \models \Diamond\phi$, we have $x \in \mathcal{C}(\llbracket\phi\rrbracket^{\mathcal{X}})$. Fix K a simplicial complex coherent with $\mathcal{X}$. By Lemma 11 $\llbracket\phi\rrbracket^{\mathcal{X}}$ is a union of cells in $\widetilde{\mathrm{K}}$, and so $\mathcal{C}(\llbracket\phi\rrbracket^{\mathcal{X}}) = \bigcup\{\mathcal{C}(\widetilde{\sigma}) \mid \widetilde{\sigma} \subseteq \llbracket\phi\rrbracket^{\mathcal{X}}\}$. In particular, $x \in \mathcal{C}(\widetilde{\sigma}) = \sigma$ for one of these cells. Fix an arbitrary element $y \in \widetilde{\sigma}$—recall that $\widetilde{\sigma}$ is nonempty whenever σ is nonempty. By Proposition A.2 (note that the names x and y are inverted in the statement of the proposition) there is a path π with $\pi(0) = y$, $\pi(1) = x$, and $\pi([0,1)) \subseteq \widetilde{\sigma} \subseteq \llbracket\phi\rrbracket^{\mathcal{X}}$. The "converse" path $\pi'(i) := \pi(1-i)$ witnesses that $\mathcal{X}, x \models \gamma(\phi, \top)$.

Proof (Lemma 15). The right-to-left direction is trivial, so we focus on the left-to-right direction. We are going to prove the result by induction on the cardinality of Π.

Base case: If $\#\Pi = 1$, then any two points $x, y \in \bigcup\Pi$ belong to the same cell $\widetilde{\sigma}$, and since cells are convex there is a linear path (thus piecewise linear) connecting x and y.

Inductive step: Suppose that $\#\Pi = n+1$ and that the result holds for sets with lower cardinality. Consider two points $x, z \in \bigcup\Pi$ and call $\widetilde{\sigma}$ the cell containing x. If $z \in \widetilde{\sigma}$ too we can reason as in the base case, so we can assume otherwise. Define Z to be the connected component of $\bigcup(\Pi \setminus \{\widetilde{\sigma}\})$ containing z. Z is itself the union of a set of cells Π'—since cells are connected. As $\#\Pi' < \#\Pi$, by inductive hypothesis Z is PL-connected. Notice that $Y := \widetilde{\sigma} \cup Z$ is connected, for otherwise Z would be disconnected from $\bigcup\Pi \setminus Z$, against the initial assumption that $\bigcup\Pi$ is connected.

$\mathcal{C}(\widetilde{\sigma}) = \sigma$ and $\mathcal{C}(Z)$ are closed sets whose union covers Y. If the intersection $Y \cap \sigma \cap \mathcal{C}(Z)$ were empty, then $Y \cap \sigma$ and $Y \cap \mathcal{C}(Z)$ would disconnect Y. So there must be a point $y \in Y \cap \sigma \cap \mathcal{C}(Z)$. By Proposition A.2 there exists a linear (thus PL) path π_1 connecting $x \in \widetilde{\sigma}$ and $y \in \sigma$, and with $\pi_1([0,1)) \subseteq \widetilde{\sigma} \subseteq Y$. Moreover, since $y \in \mathcal{C}(Z) = \mathcal{C}(\bigcup\Pi')$, there exists a cell $\widetilde{\tau} \in \Pi'$ such that $y \in \mathcal{C}(\widetilde{\tau}) = \tau$. So, again by Proposition A.2 (with w playing the role of x in the statement of the proposition), there exists a linear path π_2 from $y \in \tau$ to an arbitrary point $w \in \widetilde{\tau}$, fully contained in Y. Finally, since Z is PL-connected, there exists a PL-path π_3 from w to z in $Z \subseteq Y$. By concatenating π_1, π_2 and π_3 we obtain a PL-path connecting x and z contained in $Y \subseteq \bigcup\Pi$, as desired.

Proof (Lemma 16). The right-to-left implication follows trivially from the semantics clauses of the reachability operators, so we focus on the left-to-right implications. Fix a simplicial complex K coherent with $\mathcal{X}$. Suppose that there exists a path $\pi : [0,1] \to P$ such that $\pi(0) = x$, $\pi((0,1)) \subseteq \llbracket\phi\rrbracket^{\mathcal{X}}$ and $\pi(1) \in \llbracket\psi\rrbracket^{\mathcal{X}}$. Consider the set of cells $\Pi := \{\widetilde{\sigma} \in \widetilde{\mathrm{K}} \mid \pi((0,1)) \cap \widetilde{\sigma} \neq \emptyset\}$. Clearly $\bigcup\Pi$ is connected, and so by Lemma 15 it is also PL-connected. Moreover $x, \pi(1) \in \mathcal{C}(\bigcup\Pi)$.

Since $x \in \mathcal{C}(\bigcup \Pi)$, there exists a cell $\widetilde{\sigma} \in \Pi$ such that $x \in \mathcal{C}(\widetilde{\sigma}) = \sigma$. So by Proposition A.2 there exists a linear path π_1 from x to a point (arbitrarily chosen) $y \in \widetilde{\sigma} \subseteq \bigcup \Pi$ such that $\pi_1((0,1)) \subseteq \widetilde{\sigma}$ (note that the names x and y are inverted in the statement of the proposition). By a similar argument, there exists a linear path π_3 from a point $z \in \bigcup \Pi$ to $\pi(1) \in [\![\psi]\!]^{\mathcal{X}}$, such that $\pi_3((0,1)) \subseteq \bigcup \Pi$. Finally, since $\bigcup \Pi$ is PL-connected, there exists a PL-path π_2 from y to z completely contained in $\bigcup \Pi$. By concatenating π_1, π_2 and π_3 we obtain a PL-path π' such that $\pi'(0) = x$, $\pi'((0,1)) \subseteq \bigcup \Pi \subseteq [\![\phi]\!]^{\mathcal{X}}$ and $\pi'(1) \in [\![\psi]\!]^{\mathcal{X}}$, as desired.

Proof (Lemma 18). As cells are convex sets, the intersection between a cell and a segment is a segment. So any segment crosses each cell at most once. Observing that there are finitely many segments in a PL-path, one obtains the proof.

Proof (Theorem 21). In the proof, for S a set of logically equivalent points, we call "characteristic formula" of S the characteristic formula of the equivalence class that includes S; similarly, we also speak of the "characteristic formula" of a point x.

Consider two points x and y, with $x \equiv y$. Let us look at the conditions of Definition 20. First observe that, since ϕ can be an atomic proposition symbol, Condition 1 holds. We only prove Condition 2, as the proof of Condition 3 follows the same pattern.

Equivalently, we shall prove, by induction on k, the following statement: for each $k \geq 1$, for each pair x, y with $x \equiv y$, for each path π_x with $\pi_x(0) = x$, points $s_0, \ldots, s_k$, and cells $\widetilde{\sigma_1}, \ldots, \widetilde{\sigma_k}$ making π_x a simplicial path according to Definition 17, there is a simplicial path π_y with $\pi_y(0) = y$ and $\pi_x \hat{\equiv} \pi_y$.

To ease readability, below, given the data above, we let $\phi_1, \ldots, \phi_k$ be the characteristic formulas of the sets $\pi_x((s_0, s_1)), \ldots, \pi_x((s_{k-1}, s_k))$ respectively. Moreover, let $\phi'_0, \ldots, \phi'_k$ be the characteristic formulas of the points $\pi_x(s_0), \ldots, \pi_x(s_k)$ respectively.

Next, the proof proceeds by induction on k.

For $k = 1$, observe that, by Definition 8, we have $\mathcal{X}, x \vDash \gamma(\phi_1, \phi'_1)$. By $x \equiv y$, we have $\mathcal{X}, y \vDash \gamma(\phi_1, \phi'_1)$. By Lemma 16, there is a *piecewise linear* path π_y with $\pi_y(0) = y$, $\pi_y((0,1)) \subseteq [\![\phi_1]\!]^{\mathcal{X}}$, and $\pi_y(1) \in [\![\phi'_1]\!]^{\mathcal{X}}$. By Lemma 18, π_y is simplicial. Note that, since all the ϕ_i and ϕ'_i are characteristic formulas of equivalence classes, all the points in $\pi_x((0,1)) \cup \pi_y((0,1))$ are logically equivalent, and $\pi_x(1) \equiv \pi_y(1)$. Therefore, $\pi_x \hat{\equiv} \pi_y$.

For $k > 1$, consider the sub-paths $\pi_x^a = {\pi_x}^{[0,s_1]}$ and $\pi_x^b = {\pi_x}^{[s_1,1]}$. By Lemma A.4, both sub-paths are simplicial. By the previous case, there is a simplicial path π_y^a with $\pi_x^a \hat{\equiv} \pi_y^a$. In particular, we have $\pi_x^a(1) \equiv \pi_y^a(1)$. Noting that $\pi_x^a(1) = \pi_x(s_1) = \pi_x^b(0)$, we now apply the inductive hypothesis to the points $\pi_x^b(0)$, $\pi_y^a(1)$ and the simplicial path π_x^b, obtaining the simplicial path π_y^b with $\pi_y^b(0) = \pi_y^a(1)$ and $\pi_x^b \hat{\equiv} \pi_y^b$. Let $\pi_y = \pi_y^a; \pi_y^b$. By Lemma A.4, π_y is simplicial, and we have $\pi_x = \pi_x^a; \pi_x^b \hat{\equiv} \pi_y^a; \pi_y^b = \pi_y$, proving the thesis.

Proof (Theorem 22). Given a simplicial bisimulation $\sim$, we need to show that for every formula ϕ and all points x and y, if $x \sim y$ then $\mathcal{X}, x \models \phi \iff \mathcal{X}, y \models \phi$.

We use induction on the structure of ϕ. The cases for atomic propositions and Boolean operations are trivial, thus omitted. The case for the $\Box$ operator is also omitted, as $\Box$ is derived from γ by Theorem 13.
Suppose $x \sim y$ and $\mathcal{X}, x \vDash \phi = \gamma(\phi_1, \phi_2)$. By Lemma 16, there is a piecewise linear path π_x with $\pi_x(0) = x$, $\pi_x((0,1)) \subseteq [\![\phi_1]\!]^{\mathcal{X}}$, and $\pi_x(1) \in [\![\phi_2]\!]^{\mathcal{X}}$. By Lemma 18, π_x is simplicial. By Condition 2 of Definition 20, and the fact that $x \sim y$, there is a simplicial path π_y with $\pi_y(0) = y$, such that $\pi_x \hat{\sim} \pi_y$. For each $r \in (0,1)$, we have $\pi_x(r) \sim \pi_y(r)$ and $\mathcal{X}, \pi_x(r) \models \phi_1$. By induction hypothesis, $\mathcal{X}, \pi_y(r) \vDash \phi_1$, therefore $\pi_y((0,1)) \subseteq [\![\phi_1]\!]^{\mathcal{X}}$. We also have $\mathcal{X}, \pi_x(1) \vDash \phi_2$ and $\pi_x(1) \sim \pi_y(1)$. Thus, by induction hypothesis, $\pi_y(1) \in [\![\phi_2]\!]^{\mathcal{X}}$. Summing up, we have $\mathcal{X}, y \vDash \gamma(\phi_1, \phi_2)$, concluding one direction of our proof. Next, we should prove that conversely, if $\mathcal{X}, y \vDash \gamma(\phi_1, \phi_2)$ then $\mathcal{X}, x \vDash \gamma(\phi_1, \phi_2)$. The argument is similar to the other case, using Condition 3 of Definition 20 instead of Condition 2.

Proof (Theorem 29). Notice that by Lemma 11 the left side of the bi-implication is equivalent to $\widetilde{\sigma} \subseteq [\![\phi]\!]^{\mathcal{X}}$. We proceed by induction on the structure of the formula; the only non trivial cases are when the formula is of the form $\Box\psi$ and when the formula is of the form $\gamma(\phi, \psi)$.

Case $\Box\psi$: We have that $\mathcal{X}, x \vDash \Box\psi$ iff $x \notin \mathcal{C}([\![\neg\psi]\!]^{\mathcal{X}})$. By Lemma 11, $[\![\neg\psi]\!]^{\mathcal{X}}$ is a finite union of cells, so, by Lemma A.3, the condition is equivalent to $x \notin \bigcup\{\tau \mid \widetilde{\tau} \subseteq [\![\neg\psi]\!]^{\mathcal{X}}\}$; and again, by Lemma A.3, it is equivalent to $\forall \tau$ s.t. $\widetilde{\tau} \subseteq [\![\neg\psi]\!]^{\mathcal{X}} . \sigma \not\preceq \tau$. Again by Lemma 11, for every $\tau \in \mathrm{K}$ we have that if $\sigma \preceq \tau$ then $\widetilde{\tau} \subseteq [\![\psi]\!]^{\mathcal{X}}$, which by the inductive hypothesis amounts to $\mathcal{M}(\mathcal{X}), \sigma \vDash \Box\psi$. Note that we have equivalences in both directions of the proof, so the bi-implication is obtained "for free" for this part.

Case $\gamma(\phi, \psi)$: Firstly, suppose that $\mathcal{M}(\mathcal{X}), \sigma \vDash \gamma(\phi, \psi)$. Then there exists a $\pm$-path $\pi : \{0, \ldots, k\} \xrightarrow{\pm} S$ such that $\pi(0) = \sigma$, $\pi(\{1, \ldots, k-1\}) \subseteq [\![\phi]\!]^{\mathcal{M}}$ and $\pi(k) \in [\![\psi]\!]^{\mathcal{M}}$. By inductive hypothesis, $\widetilde{\pi}(j) \subseteq [\![\phi]\!]^{\mathcal{X}}$ for $j \in \{1, \ldots, k-1\}$ and $\widetilde{\pi}(k) \subseteq [\![\psi]\!]^{\mathcal{X}}$.

For any two simplexes $\sigma \preceq \tau$, we have $b_\sigma \in \widetilde{\sigma} \subseteq \tau$ (recall that b_σ indicates the *barycentre* of σ). So by Proposition A.2 (where b_σ and b_τ play the role of y and x respectively in the statement of the proposition) there exists a linear path connecting $b_\sigma \in \tau$ to $b_\tau \in \widetilde{\tau}$ and mapping the interval $(0,1)$ to $\widetilde{\tau}$. Using this fact, for every $i = 1, \ldots k$ we can find a linear path π'_i connecting $b_{\pi(i-1)}$ to $b_{\pi(i)}$ with $\pi'_i((0,1)) \subseteq \widetilde{\pi}(i-1)$ or $\pi'_i((0,1)) \subseteq \widetilde{\pi}(i)$—depending whether $\pi(i-1) \preceq \pi(i)$ or $\pi(i) \preceq \pi(i-1)$. Concatenating these paths, we obtain a path $\pi' := \pi'_1; \ldots; \pi'_k$ from $b_{\pi(0)}$ to $b_{\pi(k)} \in [\![\psi]\!]^{\mathcal{X}}$ such that $\pi'((0,1)) \subseteq \widetilde{\pi}(1) \cup \cdots \cup \widetilde{\pi}(k-1) \subseteq [\![\phi]\!]^{\mathcal{X}}$. This shows that $\mathcal{X}, b_{\pi(0)} \vDash \gamma(\phi, \psi)$, and so by Lemma 11 since $x, b_\pi(0) \in \widetilde{\sigma}$ we have $\mathcal{X}, x \vDash \gamma(\phi, \psi)$.

Conversely, assume that $\mathcal{X}, x \vDash \gamma(\phi, \psi)$, which by Lemma 16 amounts to the existence of a PL-path $\pi : [0,1] \xrightarrow{PL} P$ such that $\pi(0) = x$, $\pi(1) \in [\![\psi]\!]^{\mathcal{X}}$ and $\pi((0,1)) \in [\![\phi]\!]^{\mathcal{X}}$. π is a simplicial path by Lemma 18, so there exist points $s_0 = 0 < s_1 < \cdots < s_l = 1$ such that each $\pi((s_i, s_{i+1}))$ is fully contained

in a single cell. In particular, we can find cells $\widetilde{\tau}_1, \ldots, \widetilde{\tau}_l, \widetilde{\tau}'_0, \ldots, \widetilde{\tau}'_l$ such that $\pi((s_{j-1}, s_j)) \subseteq \widetilde{\tau}_j$ for $j = 1, \ldots, l$, and such that $\pi(s_j) \in \widetilde{\tau}'_j$ for $j = 0, \ldots, l$. Notice that $\widetilde{\tau}'_0 = \widetilde{\sigma}$, $\widetilde{\tau}'_l \subseteq [\![\psi]\!]^{\mathcal{X}}$ (since $\pi(1) \in \widetilde{\tau}'_l$) and $\widetilde{\tau}_i, \widetilde{\tau}'_j \subseteq [\![\phi]\!]^{\mathcal{X}}$ for $i = 1, \ldots, l$ and $j = 1, \ldots, l-1$ (since they contain points in $[\![\phi]\!]^{\mathcal{X}}$).

Observe that $\pi(s_j) \in \widetilde{\tau}'_j \cap \mathcal{C}(\widetilde{\tau}_{j+1})$ for $j = 0, \ldots, l-1$, and so by Lemma A.3 we have $\tau'_j \preceq \tau_{j+1}$. With a similar argument, we also have $\tau'_j \preceq \tau_j$ for $j = 1, \ldots, l$. Rewriting the previous conditions we have $\tau'_0 \preceq \tau_1 \succeq \tau'_1 \preceq \tau_2 \succeq \ldots \succeq \tau'_l$, which by definition means that the sequence $\langle \tau'_0, \tau_1, \tau'_1, \tau_2, \tau'_2, \ldots, \tau_l, \tau'_l \rangle$ is a $\pm$-path of $\mathcal{M}(\mathcal{X})$. By previous considerations together with the inductive hypothesis applied to ϕ and ψ, we also have that $\tau'_0 = \sigma$, that $\tau'_l \in [\![\psi]\!]^{\mathcal{M}}$ and that $\tau_i, \tau'_j \in [\![\phi]\!]^{\mathcal{M}}$ for $i = 1, \ldots, l$ and $j = 1, \ldots, l-1$. Thus we have $\mathcal{M}(\mathcal{X}), \sigma \vDash \gamma(\phi, \psi)$, as desired.

B Algorithms, pseudo-code, correctness, complexity

Currently PolyLogicA represents a polyhedral model $\mathcal{X}$ through an explicit encoding of the Kripke model from Definition 26. The latter is stored as a graph having the simplexes as nodes and with $\preceq$ as the edge relation. The current implementation stores the out-neighborhood $\texttt{out}(\sigma) = \{\tau \mid \sigma \preceq \tau\}$ and the in-neighborhood $\texttt{in}(\sigma) = \{\tau \mid \sigma \succeq \tau\}$ of each node σ in two separate arrays, allowing access in constant time to these sets.

In what follows, we indicate with n the number of simplexes and with d the dimension of K, that is, the maximum dimension of a simplex in K. Indeed, the number of nodes of the encoding is n. Moreover, since each simplex σ has at most d vertices and the faces of σ are generated by subsets of vertices, the cardinality of each out-neighborhood is at most 2^d, and so the total number of edges is at most $n \cdot 2^d$. We let N be the total size N of the Kripke structure (sum of the number of nodes and edges), which is therefore in $\mathcal{O}(n \cdot 2^{d+1})$.

The semantics of the reachability operator, as of Definition 28, is computed via a variant of the flooding procedure already employed in [21,19,31], retaining its asymptotic complexity (linear in N). The pseudo-code is reported in Figure 7. Therein, for brevity, we call "good" a $\pm$-path witnessing the formula $\gamma(\phi, \psi)$. We provide a short proof of the correctness of the algorithm.

Proof (Correctness, sketch). To be consistent with the comments in the pseudo-code, we keep calling a $\pm$-path $\pi : \{0, \ldots, k\} \to \mathrm{K}$ witnessing the satisfaction of $\gamma(\phi, \psi)$ a "good" path. First, notice that we can divide a good path in three parts: the initial point $\pi(0)$, the central segment $\pi(\{1, \ldots, k-1\})$ satisfying ϕ and the final point $\pi(k)$ satisfying ψ. To compute the set $[\![\gamma(\phi, \psi)]\!]^{\mathcal{M}}$ we work "backwards". First, we compute the set $C := [\![\phi]\!]^{\mathcal{M}} \cap \texttt{out}([\![\psi]\!]^{\mathcal{M}})$ (the simplexes of the form $\pi(k-1)$ for some good path π). Then we use a standard flooding procedure to collect the nodes of the graph that are connected to C via a non-directed path passing only through $[\![\phi]\!]^{\mathcal{M}}$ (these are the simplexes of the form $\pi(1)$ for some good path π). Finally we compute the set $[\![\gamma(\phi, \psi)]\!]^{\mathcal{M}} = \texttt{in}(D)$ (the simplices of the form $\pi(0)$ for some good path π).

```
1   Input: [[φ]]^M, [[ψ]]^M sets of nodes of a reflexive, transitive graph
2   Output: [[γ(φ,ψ)]]^M
3
4   // frontier: the points queued for the next iteration.
        ↪ Initialized with the next to last points of a good path.
5   let frontier = [[φ]]^M ∩ out([[ψ]]^M)
6
7   // flooded: all the points that are in the middle of a good path.
8   let flooded = frontier
9
10  while frontier≠∅:
11      let σ = frontier.pop()
12      for every τ ∈ in(σ) ∪ out(σ):
13          if τ ∉flooded and τ ∈ [[φ]]^M:
14              frontier.add(τ)
15              flooded.add(τ)
16
17  // result: the starting points of a good path.
18  let result = in(flooded)
19
20  return result
```

Fig. 7. Pseudo-code for model checking the reachability operator.

The code is divided in three parts, following the three steps described above: *initialization* (lines 4-8), *flooding* (lines 10-15) and *finalization* (lines 17-20). In the *initialization* we define the sets of simplexes `frontier` and `flooded`, which will be later used by the flooding procedure. Both sets are initialized using $C = [\![\phi]\!]^{\mathcal{M}} \cap \mathtt{out}([\![\psi]\!]^{\mathcal{M}})$. The *flooding* part is quite standard; it is used to compute D starting from C. At the end of this step, the value of D is stored in the variable `flooded`. Finally, in the *finalization* we return the set $\mathtt{in}(\mathtt{flooded}) = [\![\gamma(\phi,\psi)]\!]^{\mathcal{M}}$, which is the desired output of the algorithm.

The complexity of the implementation of the pseudo-code above in PolyLogicA is $\mathcal{O}(N)$—notice that all the set-theoretic operations and the flooding procedure are linear in the number of nodes and edges. The computation of the Boolean operators are also linear in N. Therefore, the asymptotic complexity of the currently implemented model checking algorithm is in $\mathcal{O}(N \cdot h)$, where h is the number of subformulas of the formula to be checked. Once d is fixed (as in the case of 3D meshes, where the "exponential" contribution of d is negligible), this becomes $\mathcal{O}(n \cdot h)$.

Next, we spend some words on the complexity of the encoding, which is also part of our tool. In the current prototype, the input is described by a list of n simplexes with maximum dimension d, each one being represented by a list of vertices. To compute the Kripke frame of Definition 26 from this description, the tool performs an explicit enumeration of the subsets of each simplex, building the arrays of out- and in-neighborhood incrementally. This results in a time complexity in $\mathcal{O}(N)$, which becomes $\mathcal{O}(n)$ once the dimension is fixed. Therefore, for d fixed, the total complexity (encoding plus model checking) is in $\mathcal{O}(n \cdot h)$.