Dynamic logics of polyhedra and their application in 3D modeling

MSc Thesis *(Afstudeerscriptie)*

written by

Kirill Kopnev
(born 23.11.1999 in Moscow, Russia)

under the supervision of **Nick Bezhanishvili** and **Vincenzo Ciancia**, and submitted to the Examinations Board in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense:	**Members of the Thesis Committee:**
25.08.2023	dr. Balder ten Cate (chair)
	dr. David Gabelaia
	dr. Alexandru Baltag
	dr. Nick Bezhanishvili (supervisor)
	dr. Vincenzo Ciancia (supervisor)

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Contents

Abstract

This thesis investigates dynamic polyhedra. We establish the connection between models of dynamic polyhedra and dynamic Kripke frames. Based on this result, we introduce a model checker for dynamic models. We demonstrate the application of `PolyLogicA` model checker on a real-world example from the architectural domain. Several novel theoretical results are obtained, including the Hennessy-Milner theorem for the language of basic modal logic extended by reachability modality γ and polyhedral completeness for two dynamic logics. The work contributes to the field of spatial reasoning, bridging theoretical and practical aspects in spatial model checking for dynamic objects.

Chapter 1

Introduction

This thesis lies at the intersection of two research strains in spatial reasoning. Initially, they had different motivations. The first was driven more by theoretical interest and the other by applications. The exchange of ideas makes it impossible to separate the two, combining them into one remarkable line of research called polyhedral semantics. We will discuss the story of this direction below and finish this introduction by explaining what role our thesis fulfils in this story.

History The origins of the connection between spatial structures and logic lie in the work of McKinsey and Tarski [Tar39] and [MT44]. This work is renowned for introducing *topological semantics*, a novel approach to modal logic at that time. Logical formulas in this setting are interpreted on subsets of topological spaces, while modal operators "box" and "diamond" are interpreted as interior and closure of the topology, respectively. Topological semantics has proved to be productive and fruitful research in spatial reasoning for decades, as seen in an overview article by van Bentham and Bezhanishvili [BB07]. In recent years, polyhedral semantics has gained increasing interest (see, for example, [Bez+18] and [Ada+22]). In this semantics, we consider only polyhedra spaces instead of any topological space. The interpretation of formulas, in this case, can be defined on the subpolyhedra of a given polyhedron. Within this line of research, interesting completeness results have been obtained; for instance, the logic of all polyhedra is the well-known modal system **S4.Grz** [Bez+18]. Working with polyhedra has many advantages over topological spaces because polyhedrons can represent real objects. We can approximate the real objects which can be represented as a polyhedron using the notion of *simplicial complex*.

For this reason, polyhedral semantics is also interesting for *spatial model checking* [Cia+14]. In the classic case, model checking methods are used to verify the properties of information technology systems to identify and eliminate bugs and errors [BK08]. For this purpose, it is necessary to formulate the mathematical model of the system and a formal language. The language is used to formulate the properties of the system. The model checking algorithm is given a mathematical model and a formula as input. It checks whether the property expressed by the formula is true either globally or locally in the model. For example, if we work with a concurrent system, the property to be checked might be "the system can never reach a deadlock." Polyhedral semantics gives us both the tools for the mathematical modeling of 2D and 3D images and the formal language for describing their spatial properties. That is why it becomes a good candidate for spatial model checking. A project that revealed the possibilities of polyhedral semantics in 3D image analysis is PolyLogicA[1] [Bez+22]. A new bimodal operator γ with the semantics "an area A is reachable through area B" is introduced. Adding γ to the basic modal language makes possible to identify a rich variety of spatial properties of 3D models. Consider an example of a building. Using γ we are able to formulate a property "an exit is reachable from room A". The project `PolyLogicA` resulted in a prototype for model checking on 3D models, which was tested on an existing 3D model from the medical domain. We discuss this prototype in Chapter 4.

Our study We will call *static polyhedral semantics* the previous research in polyhedral semantics , i.e. semantics that describes static objects. Even though this field still contains some open problems, it can be considered as well-studied. However, the world is full of dynamic objects, so it is pretty natural to start a new research direction in polyhedral semantics, namely in the direction of *dynamic polyhedral semantics*. The study of dynamic systems and their logic has been a rich research field of spatial reasoning [ADN97],[Kon+07], [KM07]. We continue this line of research, but instead of topological spaces, we work with polyhedra and dynamics on them. As well as in the case of static polyhedral semantics, we can take a simplicial complex that approximates some dynamic object and investigate spatial-temporal properties of it. Consider, for instance, some emerging situations (e.g. fire spreading in the building). Then at different stages of time, the properties of some regions in the building might change. For example, at the first moment, it would still be possible to reach the exit from room A, but at the second moment, it may become impossible due to the developed conditions. To be able to identify the regions that will be safe at the next moment, we have to formalize this process with some model $\mathcal{M}$, formulate the logical formula φ for extracting all the safe points of the model in the future moment, and input them into the model checker. The first part of our thesis is devoted to these questions, i.e. the definition of dynamic semantics for

[1] https://github.com/vincenzoml/VoxLogicA/tree/polyhedra

polyhedra and providing a model checker for checking spatial-temporal properties of dynamic models. In the second part of the thesis, we abstract from the models and concentrate solely on the dynamic systems and their logic.

To use the model checking program on an object, we have to encode this object in a computer. Thus, we will assume that we work with 3D models and dynamic 3D models. We list the main new results and outcomes of this thesis:

1. An analogue of the Hennessy-Milner theorem for bisimulation on Kripke frames with modal operator γ is proved (Corollary2.4.11);

2. A connection between the truth of a formula on a dynamic polyhedra and a dynamic Kripke frame is established (Theorems 3.2.11 and 3.3.14);

3. Correctness and complexity bound of novel model checking algorithms for dynamic models (Algorithm 1 and Algorithm 2);

4. A utilization of the PolyLogicA prototype on a new example from the architectural domain (Figure 4.4);

5. A conceptual outline for constructing the model checker for dynamic 3D models (Section 4.2);

6. A polyhedral completeness result for two logics of dynamic systems (Theorem 5.3.19).

Structure of the thesis In Chapter 2, we provide all the preliminaries. We present a framework for model checking on the static case of 3D models. Most of the results are taken from [Bez+22]. The original result obtained in this chapter is the Hennessy-Milner theorem for bisimulation on Kripke models in a new modal language with γ. In Chapter 3, we define a semantics of dynamic systems and provide all the necessary tools for doing model checking on dynamic 3D models. Then, in Chapter 4, we present a model checking algorithm for our models and address how a prototype of such a dynamic model checker should be built. In the same chapter, we present an application of a model checker on a new example from the architectural domain. Finally, in Chapter 5, we define two dynamic logics $\mathcal{DPL}$ and $\mathcal{DRL}$ and prove their polyhedral completeness.

Chapter 2

Static polyhedral semantics

The main objective of this chapter is to prepare the ground for the following research on model checking of dynamic 3D models and the logic of dynamic systems. In the first section, we introduce the concepts related to polyhedra. The second section defines a language with new reachability modality γ and its semantics on polyhedra. The third section builds a connection between polyhedra and partial orders, which we call Kripke frames. This section is finalized with the theorem that establishes the correspondence between the truth of a formula on polyhedra and the truth of its encoding, which is a finite Kripke model. This result shows that the problem of checking the truth of a formula in a polyhedron can be reduced to the problem of checking the truth of a formula on a Kripke frame. Therefore, defining a model checking algorithm on the Kripke model is sufficient. Finally, in the last section of this chapter, we investigate the notion of bisimulation on Kripke frames for the new language we work with. This research direction is vital for future research on optimizing the model checking algorithm. Sections 1, 2 and 3 contain mostly the definitions and results from [Bez+22]; the proofs for most lemmas and theorems can also be found there. The last section includes a new result on the Hennessy-Milner property for Kripke frames in the basic modal language with γ.

2.1 Polyhedra

In this section, we recall the definition of a polyhedron, simplexes, simplicial complexes, and triangulation of a polyhedron.

Definition 2.1.1 (Polyhedron)**.**

1. An *affine combination* of $v_0, \ldots, v_d \in \mathbb{R}^n$ is a point $\lambda_0 v_0 + \ldots + \lambda_d v_d$, specified by some $\lambda_0, \ldots, \lambda_d \in \mathbb{R}$ such that $\lambda_0 + \cdots + \lambda_d = 1$.

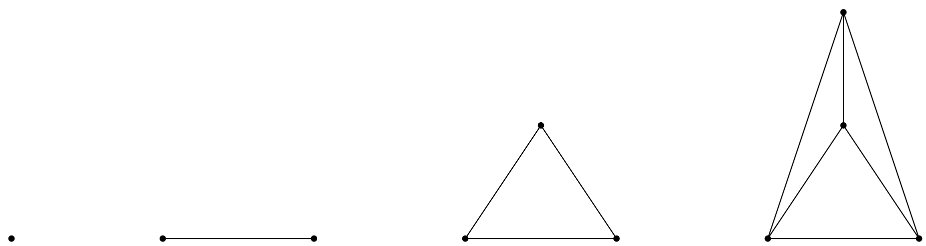

Figure 2.1: 0, 1, 2 and 3–dimensional simplexes.

2. A *convex combination* is an affine combination in which additionally each $\lambda_i \geqslant 0$.

3. Given a set $S \subseteq \mathbb{R}^n$, its *convex hull* $Conv(S)$ is the collection of convex combinations of its elements.

4. A subspace $S \subseteq \mathbb{R}^n$ is *convex* if $Conv(S) = S$.

5. A *polytope* is the convex hull of a finite set.

6. A *polyhedron* P in $\mathbb{R}^n$ is a set that can be expressed as the finite union of polytopes.

Definition 2.1.2. Points $v_0, \dots, v_d \in \mathbb{R}^n$ are *affinely independent (a.i.)* if $v_1 - v_0, \dots, v_d - v_{d-1}$ are linearly independent.

Definition 2.1.3. A *d-simplex σ* is the convex hull of affinely independent points $v_0, \dots, v_d \in \mathbb{R}^n$.

The number d is called the dimension of σ and $v_0, \dots, v_d$ are called its vertices.

Since any subset of a.i. points is also a set of a.i. points, thus, its convex hull is a simplex τ. We call such τ a *face* of σ (in symbols $\tau \preccurlyeq \sigma$), and call it a *proper face* if $\tau \neq \varnothing$ and $\tau \neq \sigma$.

Simplexes can be seen as the simplest linear convex bounded shapes. A 2-dimensional simplex is a *triangle*; a 3-dimensional simplex is a *tetrahedron* etc. See figure 2.1 for examples of some simplexes. Note that the two-dimensional faces of a tetrahedron are triangles (2-dimensional simplexes), which faces are line segments (1-dimensional simplexes) and points (0-dimensional simplexes).

For our work, it is important to have not only simplexes but also the "internal part" of them.

Definition 2.1.4 (Relative interior)**.** The *relative interior* of a simplex σ is the set:

$$\widetilde{\sigma} := \left\{ \sum_{i=0}^{d} \lambda_i v_i \,\middle|\, \forall i : \lambda_i \in (0,1] \text{ and } \sum_{i=0}^{d} \lambda_i = 1 \right\}.$$

Note that for any non-empty simplex σ, its relative interior will also be non-empty. For a 0-simplex σ, $\widetilde{\sigma} = \sigma$. Observe that a simplex σ can be represented as the union of the relative interiors of its faces: $\sigma = \bigcup\{\widetilde{\tau} \mid \tau \preccurlyeq \sigma\}$.

We assume that the reader is familiar with the closure and interior operators in topological spaces. We denote the *closure* of a set A by $\mathsf{Cl}(A)$ and its *interior* by $\mathsf{Int}(A)$.

Definition 2.1.5. A *simplicial complex* K is a finite set of simplexes of $\mathbb{R}^n$ such that:

1. If $\sigma \in K$ and $\tau \preccurlyeq \sigma$, then $\tau \in K$ (*downward closeness*);

2. If $\sigma, \tau \in K$, then $\sigma \cap \tau$ is a face of σ and τ (*intersection property*).

The *support* of a simplicial complex K is $|K| := \bigcup_{i=1}^{n} \sigma_i$. Note that, by definition, it is a polyhedron. We say that K is a *triangulation* of the polyhedron $|K|$. A subcomplex K' of K is a subset which itself is a simplicial complex.

Lemma 2.1.6. Each point of $|K|$ belongs to the relative interior of exactly one non-empty simplex in K. That is, $\widetilde{K} := \{\widetilde{\sigma} \mid \sigma \in K \backslash \{\varnothing\}\}$ is a partition of $|K|$.

Proof. See [Bez+22, Lemma 2.4]. □

We call $\widetilde{K}$ a *simplicial partition* of a polyhedron $|K|$. The above result justifies the introduction of the following notation: for $x \in |K|$, let us denote by $\widetilde{\sigma}^x$ the unique simplex, such that $x \in \widetilde{\sigma}^x$. For a given polyhedron P, we will call P' a *subpolyhedron* if $P' \subseteq P$ and P' is a polyhedron. We denote the set of all subpolyhedra of P by $Sub(P)$.

Lemma 2.1.7. Any polyhedron P admits a triangulation that simultaneously triangulates each of any fixed finite set of subpolyhedra. That is, for a collection of polyhedra $P, Q_1, \ldots, Q_m$ with $Q_i \subseteq P$, there is a triangulation Σ of P such that K_{Q_i} triangulates Q_i for each i.

Proof. See [RS12, Theorem 2.11]. □

From now till the end of this section, we fix a polyhedron P and its triangulation K.

The following lemma shows that every subset of a simplicial complex K generates a subpolyhedron of P.

Lemma 2.1.8. If $C \subseteq K$, then $\bigcup C \in Sub(P)$.

Proof. Take $C \subseteq K$. Then define $D = \{\sigma \mid \exists \tau \in C : \sigma \preccurlyeq \tau\}$. By definition, D is downward closed and $D \subseteq K$, D also has the intersection property. Thus, D is a simplicial complex. Observe that $|D| = \bigcup C$, since $\sigma \preccurlyeq \tau$ iff $\sigma \subseteq \tau$. Therefore, $\bigcup C \in Sub(P)$. $\square$

The last essential definitions for further discussion are definitions of a continuous function on the topological space and topological path.

Definition 2.1.9. A function $f : \mathcal{X} \to \mathcal{X}$ on a topological space $\mathcal{X}$ is called *continuous*, whenever $f^{-1}(U)$ is an open set for every open $U \subseteq \mathcal{X}$.

Definition 2.1.10. A *topological path* in a topological space $\mathcal{X}$ is a continuous function $\pi : [0, 1] \to \mathcal{X}$, where interval $[0, 1]$ is equipped with the subspace topology of $\mathbb{R}$.

2.2 Language and semantics

Definition 2.2.1. The *modal language* $\mathcal{L}_{\Box\gamma}$ is defined using a set of proposition letters Prop, whose elements we denote by p, q, r, etc., unary modal operator $\Box$ and binary modal operator γ. The well-formed formulas φ of the language $\mathcal{L}_{\Box\gamma}$ are defined by the rule:

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

where p is an atomic proposition from Prop.

Remark 2.2.2. The basic modal language, which contains only the modality $\Box$, is quite limited in its ability to express topological properties. For example, McKinsey and Tarski showed in [MT44] that the modal logic **S4** enjoys completeness with respect to the class of all topological spaces, the real line, and the Cantor space. In other words, it does not make a distinction between them. Consequently, it is common to extend the basic language with additional modalities in order to make the language more expressive. Shehtman's work [She99] illustrates how the introduction of a universal modality allows the formulation of the connectedness of topological spaces. In our story, we also use a new modality, γ, to capture the properties of region connectedness in polyhedra.

As we will see from the semantics, the new bimodal operator γ has an analogous meaning to the modality Until in temporal modal logic [BDV01]. The difference is that, unlike Until, γ has a spatial rather than a temporal interpretation. We will refer to γ as *reachability operator*, and $\gamma(\varphi, \psi)$ is pronounced as "ψ is reachable by φ".

Now, having the language fixed, we can define the semantics of our language.

Definition 2.2.3. A tuple $\mathcal{E} = (P, K, V)$ is a *polyhedral model*, whenever P is a polyhedron, K is the triangulation of P and $V : \mathrm{Prop} \to \mathfrak{U}(K)$, where $\mathfrak{U}(K) = \{\bigcup \widetilde{K}' \mid K'\text{- subcomplex of } K\}$.

The model we defined can be treated as a general topological frame, i.e. a topological space with a restricted valuation function. In our case, we restrict our valuation to the elements of $\mathfrak{U}(K)$. And since $\mathfrak{U}(K)$ is finite, there can be only finitely many propositional letters evaluated differently.

Definition 2.2.4. Let $\mathcal{E} = (P, K, V)$ be a polyhedral model. We define inductively when a formula φ in language $\mathcal{L}_{\Box,\langle\mathrm{R}\rangle}$ is satisfied at a point x (notation: $\mathcal{E}, x \vDash \varphi$):

$$
\begin{array}{lcl}
\mathcal{E}, x \vDash p & \iff & x \in V(p) \text{ for } p \in \mathrm{Prop} \\
\mathcal{E}, x \vDash \neg\varphi & \iff & \mathcal{E}, x \nvDash \varphi \\
\mathcal{E}, x \vDash \varphi \wedge \psi & \iff & \mathcal{E}, x \vDash \varphi \text{ and } \mathcal{E}, x \vDash \psi \\
\mathcal{E}, x \vDash \Box\varphi & \iff & x \in \mathrm{Int}\left(\llbracket\varphi\rrbracket^{\mathcal{E}}\right) \\
\mathcal{E}, x \vDash \gamma(\varphi, \psi) & \iff & \text{there exists a path } \pi \text{ such that} \\
 & & \pi(0) = x, \pi(1) \in \llbracket\psi\rrbracket^{\mathcal{E}} \text{ and } \pi((0,1)) \subseteq \llbracket\varphi\rrbracket^{\mathcal{E}}
\end{array}
$$

where the set $\llbracket\varphi\rrbracket^{\mathcal{E}}$ stands for the set of all points in the model $\mathcal{E}$, where φ is true. We will omit the superscript when its abundance leads to no ambiguity.

For the simplicity of notation, we define $\Diamond\varphi := \neg\Box\neg\varphi$. Then it is easy to see that $\llbracket\Diamond\varphi\rrbracket = \mathsf{Cl}(\llbracket\varphi\rrbracket)$.

Our next step is to show that not only are propositional letters evaluated on the elements of the set $\mathfrak{U}(K)$, but that for every formula φ the truth set $\llbracket\varphi\rrbracket$ is in $\mathfrak{U}(K)$. To formulate this result, we need to define a logical equivalence relation on a model with respect to the formulas of any modal language $\mathcal{L}_{\nabla_1,\ldots,\nabla_n}$.

Definition 2.2.5. Let $\mathcal{E} = (P, K, V)$ be a polyhedral model, and $\mathcal{L} = \mathcal{L}_{\nabla_1,\ldots,\nabla_n}$ be a modal language. *Logical equivalence* $\equiv_{\mathcal{L}}$ is the binary relation on P such that $x \equiv_{\mathcal{L}} y$ if for every formula φ in $\mathcal{L}$:

$$\mathcal{E}, x \vDash \varphi \iff \mathcal{E}, y \vDash \varphi.$$

Lemma 2.2.6. Let $\mathcal{E} = (P, K, V)$ be a polyhedral model. Then for each formula, for each cell $\widetilde{\sigma} \in \widetilde{K}$, for each points $x, y \in \widetilde{\sigma}$, we have that $x \equiv_{\mathcal{L}_{\Box\gamma}} y$.

Proof. We will provide a sketch of the proof. For the full proof, see [Bez+22, Lemma 3.6]. For the formula $\Box\varphi$, it is sufficient to show that $P\backslash\mathsf{Cl}(P\backslash[\![\varphi]\!])$ is in $\mathfrak{U}(K)$. By inductive hypothesis, $[\![\varphi]\!]$ is an element in $\mathfrak{U}(K)$. Since for every $\widetilde{\tau}$ we have $\mathsf{Cl}(\widetilde{\tau}) = \tau$, then $\mathsf{Cl}(P\backslash[\![\varphi]\!])$ is the union of elements in $\widetilde{K}$. Then it is also in $\mathfrak{U}(K)$, and its complement is in $\mathfrak{U}(K)$ as well. For formula $\gamma(\varphi, \psi)$, take the path π which starts in x, goes through $[\![\varphi]\!]$, and finishes at $[\![\psi]\!]$. The general idea is to take a simplex $\tau \ni y$, such that $\widetilde{\tau} \subseteq [\![\varphi]\!]$ with a point of the path $\pi(r) \in \widetilde{\tau}$. Then we can build a path from y to $\pi(r)$, such that $(y, \pi(r)) \subseteq \widetilde{\tau}$, and take the tail of the path π, starting from $\pi(r)$. Merging these two halves gives us the desired path from y. $\square$

We obtain the following proposition:

Proposition 2.2.7. Let $\mathcal{E} = (P, K, V)$ be a polyhedral model. Then for every formula φ in $\mathcal{L}_{\Box\gamma}$, we have $[\![\varphi]\!] \in \mathfrak{U}(K)$.

Proof. We will show that $[\![\varphi]\!] = \bigcup\limits_{x \in [\![\varphi]\!]} \widetilde{\sigma}^x$. The non-trivial direction is from right to left. If $x \in [\![\varphi]\!]$, then we have that for every $y \in \widetilde{\sigma}^x$ by Lemma 2.2.6. Hence, $\subseteq [\![\varphi]\!] \subseteq \widetilde{\sigma}^x$. $\square$

Since each formula is evaluated on the relative interiors of the simplexes in K, we do not have to consider all the points in our polyhedrons when evaluating the formula. Instead, we can look only at the elements in $\widetilde{K}$, which is essentially finite. In the next section, we will show how to go from a polyhedron to a relational structure (a finite graph), which encodes this polyhedron. Then instead of checking the truth value of a given formula on a polyhedron, we will simply run a model checking algorithm on its relational representation. Such model checking algorithms for relational structures are well established by now [Bez+22].

We finalize this section with a little investigation of the mutual definability of operators $\Box$ and γ. From the [Bez+22], we know that modality $\Box$ can be expressed using γ.

Proposition 2.2.8. For polyhedral model $\mathcal{E} = (P, K, V)$:

$$\mathcal{E}, x \vDash \Box\varphi \text{ iff } \mathcal{E}, x \vDash \neg\gamma(\neg\varphi, \top).$$

Proof. See [Bez+22, Theorem 3.8]. $\square$

However, can γ be expressed using a formula in a language with $\Box$? The answer to this question is negative. This result can be established using the theorem from topological semantics for modal logic. The following theorem can be found in [BB07, Theorem 5.4].

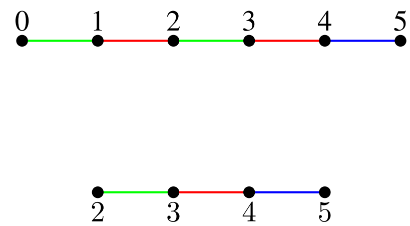

Figure 2.2: Illustration for the proof of Theorem 2.2

.

Theorem 2.2.9. If $(\mathcal{X}, V)$ is a topological model and $f : \mathcal{X} \to \mathcal{X}$ is an interior map that preserves and reflects the valuations (i.e. $x \in V(p)$ iff $f(x) \in V(p)$), then $f(x) \equiv f(y)$ iff $x \equiv y$.

However, for language with operator γ, this result does not hold.

Theorem 2.2.10. There is a polyhedral model $\mathcal{E} = (P, K, V)$ and an interior map $f : \mathcal{P} \to \mathcal{P}$ which preserves and reflects the valuations (i.e. $x \in V(p)$ iff $f(x) \in V(p)$), but it is not the case that $f(x) \equiv f(y)$ iff $x \equiv y$.

Proof. Take interval $[0, 5]$ and a map $f : [0, 5] \to [0, 5]$ between them, such that:

- $f([0, 1]) = [2, 3] = f([2, 3])$
- $f((1, 2)) = (3, 4) = f((3, 4))$
- $f([4, 5]) = [4, 5]$

and $f \restriction_{[0,1]}$, $f \restriction_{[2,3]}$, $f \restriction_{(1,2)}$, $f \restriction_{(3,4)}$ and $f \restriction_{[4,5]}$ are homeomorphisms. Take the valuation on $V : P \to P$ such that

- $V(p) = [0, 1] \cup [2, 3]$;
- $V(q) = (1, 2) \cup (3, 4)$;
- $V(r) = (4, 5)$

Then f reflect and preserves the valuations, however $\mathcal{E}, 1 \nvDash \gamma(\neg p, r)$ but $\mathcal{E}, f(1) \vDash \gamma(\neg p, r)$. Thus, $f(1) \equiv f(1)$, but $\neg 1 \equiv 3$. This is illustrated in figure 2.2. $\square$

Therefore, $\mathcal{L}_{\square\gamma}$ is more expressive than $\mathcal{L}_{\square}$.

2.3 Kripke semantics

In the previous section, we demonstrated that logic enables us to study polyhedra interpreting formulas on the triangulation of a polyhedron. However, this alone does not shed much light on how to design an algorithm that checks the truth of a formula in a polyhedron. In the realm of computer science, graph-based representations are often favoured for defining algorithms. Fortunately, in our case, we can establish a correspondence between a polyhedron and a finite partial order, which reflects the truth of all the formulas within our polyhedral model. We start with the definition of a Kripke model.

Definition 2.3.1. Let $\sqsubseteq$ be a relation on W. We say that $\sqsubseteq$ is a poset relation on W if it is reflexive, transitive and antisymmetric.

Definition 2.3.2.

1. A *Kripke frame* is a tuple $\mathcal{F} = (W, \sqsubseteq)$, where W is a set, and $\sqsubseteq$ is a poset relation on W.
2. We call $\mathcal{M} = (W, \sqsubseteq, V)$ a Kripke model, where $(W, \sqsubseteq)$ is a Kripke frame and $V : \mathrm{Prop} \to \mathcal{P}(W)$.

For ease of notation, we will sometimes write $(\mathcal{F}, V)$ instead of $(W, \sqsubseteq, V)$. We will call a Kripke frame $\mathcal{F} = (W, \sqsubseteq)$ finite, whenever W is finite.

Remark 2.3.3. Initially, the study of polyhedral semantics started from the side of intuitionistic logic. That is why only partial orders were needed as frames. And since we will consider no Kripke frames that are not posets, there is no need to define them as a set with a relation.

Definition 2.3.4. Let $\mathcal{F} = (W, \sqsubseteq, f)$ be a Kripke frame. We call $\mathcal{F}' = (W', \sqsubseteq')$ a *subframe* of $\mathcal{F}$, if $W' \subseteq W$ and $\sqsubseteq'$ is the restriction of $\sqsubseteq$ to W'.

We will now give some additional definitions and notations related to Kripke frames.

Definition 2.3.5. We call $C \subseteq W$ a *downset* if for all $x \in C$ if $y \sqsubseteq x$ for some y, then $y \in C$.

Regarding this definition, we also define an operation $\downarrow$ which we call a *closure* of a set $A \subseteq W$,

$$\downarrow A = \{x \mid \exists y \in A : x \sqsubseteq y\}.$$

Since $\sqsubseteq$ is a poset relation, we can also define the height of a frame $\mathcal{F}$ (resp. model $\mathcal{M}$).

Definition 2.3.6.

1. A *chain* in W is a set $X \subseteq W$ which as a subposet is linearly-ordered;
2. The *length* of a chain X is $|X|$;
3. Take any subframe $\mathcal{F}' \subseteq \mathcal{F}$. A chain $X \subseteq \mathcal{F}'$ is *maximal* (in $\mathcal{F}'$) if there is no chain $Y \subseteq \mathcal{F}'$ such that $X \subset Y$ (i.e. such that X is a proper subset of Y).
4. A chain X is *strict* if there are no $x < y < z$ such that $x, z \in X$ but $y \notin X$;
5. The *height* of $\mathcal{F}'$ is the element of $\mathbb{N} \cup \{\infty\}$ defined by:

$$\textbf{height}(\mathcal{F}') := \sup\{|X| - 1 \mid X \subseteq W' \text{ is a chain }\}/$$

To define the semantics of formulas in $\mathcal{L}_{\Box\gamma}$ on Kripke frames, we will introduce a suitable definition of a path on a Kripke frame that will allow us to find the logical correspondence between polyhedral models and Kripke models.

Definition 2.3.7 ($\pm$ - path)**.** Given a Kripke frame $\mathcal{F} = (W, \sqsubseteq)$, let $\sqsubseteq^{+}$ be the relation $\sqsubseteq \cup \sqsupseteq$. We say that $\pi : \{0, \ldots, k\} \to W$ is a $\pm$ – *path* if $k \geq 2$ and $\pi(0) \sqsubseteq \pi(1) \sqsubseteq^{+} \pi(2) \sqsubseteq^{+} \ldots \sqsubseteq^{+} \pi(k-1) \sqsupseteq \pi(k)$.

This notion is analogous to the one of a continuous path in a polyhedron. Having this concept, we can interpret the γ operator on our structures.

Definition 2.3.8. Let $\mathcal{M} = (W, \sqsubseteq, V)$ be a Kripke model, $x \in W$ and $\varphi \in$ S. We define recursively when a formula φ in language $\mathcal{L}_{\Box,\gamma}$ is satisfied at a point x (notation: $\mathcal{M}, x \vDash \varphi$):

$$
\begin{array}{lcl}
\mathcal{M}, x \vDash p & \iff & x \in V(p) \text{ for } p \in \mathrm{Prop} \\
\mathcal{M}, x \vDash \neg\varphi & \iff & \mathcal{M}, x \nvDash \varphi \\
\mathcal{M}, x \vDash \varphi \wedge \psi & \iff & \mathcal{M}, x \vDash \varphi \text{ and } \mathcal{M}, x \vDash \psi \\
\mathcal{M}, x \vDash \Box\varphi & \iff & \forall y : x \sqsubseteq y \Rightarrow \mathcal{M}, y \vDash \varphi \\
\mathcal{M}, x \vDash \gamma(\varphi, \psi) & \iff & \text{there exists a } \pm\text{- path } \pi : \{0, \ldots, k\} \to W \text{ such that} \\
 & & \pi(0) = x, \pi(k) \in [\![\psi]\!]^{\mathcal{M}} \text{ and } \pi(\{1, \ldots, k-1\}) \subseteq [\![\varphi]\!]^{\mathcal{M}}
\end{array}
$$

where the set $[\![\varphi]\!]^{\mathcal{M}}$ stands for the set of all points in the model $\mathcal{M}$, where φ is true. We will omit the superscript when its abundance leads to no ambiguity.

Since every polyhedral model is equipped with a triangulation K, we will use the face relation $\preccurlyeq$ on K to obtain a Kripke frame.

Definition 2.3.9. Let P be a polyhedron and K a simplicial complex. We define a *Kripke frame of* P *and* K as a pair $(\widetilde{K}, \precsim)$, such that $\precsim$ is a relation on $\widetilde{K}$ s.t.:

$$\widetilde{\sigma}_1 \precsim \widetilde{\sigma}_2 \text{ iff } \sigma_1 \preccurlyeq \sigma_2.$$

Figure 2.3 illustrates the relationship between a simplex with a trivial triangulation and its Kripke frame.

Definition 2.3.10. Let σ be a simplex. Integer d is a height of simplex σ, whenever **height**$(\downarrow\{\widetilde{\sigma}\}) = d$.

Observe that for d-simplex σ we have that **height**$(\sigma) = d$. We also have that every simplex of height d is incompatible with other simplexes of height d.

Definition 2.3.11. Let K be a simplicial complex. Then define:

$$\widetilde{K}_i = \{\widetilde{\sigma} \mid \sigma \text{ } i\text{-simplex}\}$$

We call $\widetilde{K}_i$ the i-*level* of $\widetilde{K}$.

We also present a useful lemma without proof for working with simplicial complexes.

Lemma 2.3.12. Let K be a simplicial complex and $\sigma, \tau \in K$. Then $\sigma \preceq \tau$ iff $\widetilde{\sigma} \widetilde{\preceq} \tilde{\tau}$ iff $\widetilde{\sigma} \subseteq Cl(\tilde{\tau})$ iff $\widetilde{\sigma} \cap Cl(\tilde{\tau}) \neq \varnothing$.

Proof. See [Bez+22, Lemma A.3]. □

Lemma 2.3.13. Let P be a polyhedron and K its triangulation. For C a subset of $\widetilde{K}$: C is downset iff $\bigcup C \in Sub(P)$.

Proof. ($\Rightarrow$) If $C = \{\widetilde{\sigma}_1, \ldots, \widetilde{\sigma}_n\}$ is downset, then for every $\widetilde{\sigma}_i$ we have that $\{\widetilde{\tau} \mid \widetilde{\tau} \preccurlyeq \widetilde{\sigma}_i\} \subseteq C$. Then $\sigma_i \subseteq \bigcup C$. Hence, $\bigcup C = \bigcup D$, where $D = \{\sigma \mid \widetilde{\sigma} \in C\}$. By Lemma 2.1.8 $D \in Sub(\mathrm{P})$.

($\Leftarrow$) Take $\widetilde{\tau}$ s.t. $\widetilde{\tau} \precsim \widetilde{\sigma}$ and $\widetilde{\sigma} \in C$. Then by the definition of a polyhedron, we must have that $\widetilde{\tau} \subseteq \bigcup C$. Since C consists of pairwise disjoint elements, we have that $\widetilde{\tau} \in C$. □

To define a model $\mathcal{M}(\mathcal{E})$ from a polyhedral model $\mathcal{E}$, we have to define function $V : \mathrm{Prop} \to \mathcal{P}(\widetilde{K})$.

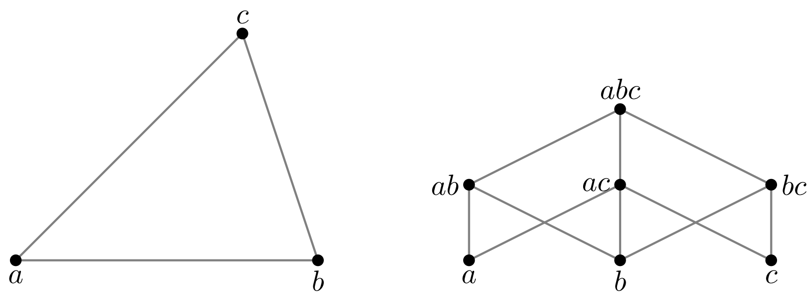

Figure 2.3: Here you can see how a simplex abc consisting of the relative interiors of abc, ac, ab, bc, a, b, c can be represented by a relational structure (which is a poset) whose points are the same relative interiors. The nodes in the bottom row represent the zero-dimensional simplexes (i.e. the vertices), the middle row the one-dimensional simplexes (i.e. the line segments) and the top row of the two-dimensional simplexes (i.e. the triangles).

Definition 2.3.14. Given a polyhedral model $\mathcal{E} = (P, K, V)$, we define its *encoding* $\mathcal{M}(\mathcal{E}) = (\widetilde{K}, \precsim, \widetilde{V})$, in the following way:

- $(\widetilde{K}, \precsim)$ is a Kripke frame of P and K from definition 2.3.9;
- $\widetilde{V}(p) = \widetilde{\sigma}$, whenever $\widetilde{\sigma} \subseteq V(p)$.

As the reader can see, the model $\mathcal{M}(\mathcal{E})$ reflects the truth of the propositional letter in $\mathcal{E}$. Thus, to check the truth of a modal formula φ on $\mathcal{E}$, we need the following theorem.

Theorem 2.3.15. Let $\mathcal{E} = (\mathcal{P}, K, V)$ be a polyhedral model. Then for every $x \in P$ and every formula φ in S we have

$$\mathcal{E}, x \vDash \varphi \iff \mathcal{M}(\mathcal{E}), \widetilde{\sigma}^x \vDash \varphi.$$

Proof. Here we will present only the sketch of the proof. The full proof can be found in [Bez+22, Theorem 4.4].

For the case $\Box\varphi$, assume that $x \notin \mathsf{Cl}(\llbracket \neg\varphi \rrbracket)$.

Then by Proposition 2.2.7 we have that $x \notin \{\tau \mid \widetilde{\tau} \subseteq \llbracket \varphi \rrbracket^{\mathcal{E}}\}$. By lemma 2.3.12 it follows that $\neg(\widetilde{\sigma}^x \precsim \widetilde{\tau})$ for every $\widetilde{\tau} \subseteq \llbracket \neg\varphi \rrbracket^{\mathcal{E}}$. So by the proposition 2.2.7 we have that if $\widetilde{\sigma}^x \precsim \widetilde{\tau}$, then $\widetilde{\tau} \subseteq \llbracket \psi \rrbracket^{\mathcal{M}(\mathcal{E})}$. In other words, $\mathcal{M}(\mathcal{E}), \widetilde{\sigma}^x \vDash \Box\varphi$. Since the equivalences hold in both directions, the result follows.

For the case $\gamma(\varphi, \psi)$ [Bez+22] first introduces the notion of piecewise linear paths (PL-paths), i.e. paths consisting of several merged intervals. With this notion it is proved $\mathcal{E}), x \vDash \gamma(\varphi, \psi)$ if there exists a PL-path π such that $\pi(x) = 0$,

$\pi((0,1)) = [\![\varphi]\!]\mathcal{E}$ and $\pi(1) \in [\![\psi]\!]\mathcal{E}$. Then it is shown that every PL-path can be divided into intervals such that each interval lies within a relative interior of some simplex. After that, the proof is straightforward since for every $\pm$-path, we can construct a PL-path in our polyhedron, and every PL-path corresponds to a $\pm$-path in $\mathcal{M}(\mathcal{E})$. $\square$

This is a key result for us. Now it is sufficient to define a model checking algorithm on arbitrary encodings since it is sufficient to check the truth of a formula on its encoding $\mathcal{M}(\mathcal{E})$ to check whether $\mathcal{E}, x \vDash \varphi$ for a polyhedral model $\mathcal{E} = (P, K, V)$. And this is a much easier task than the first one. However, we need to define the algorithm itself. We will deal with this question in chapter 4.

2.4 Simplicial Bisimulation

In [Bez+22], the authors also explore the concept of bisimulation for polyhedral models. The key idea is to identify spatially equivalent regions before applying model checking. This will lead to the improvement of the performance of geometric spatial model checking by simplifying the triangulation of the polyhedron [Cia+23]. Similar to the classical approach to research on bisimulation, the authors establish the connection between bisimulation and logical equivalence, i.e. Hennessy-Milner result for bisimulation [BDV01]. Our research follows a similar trajectory but focuses on Kripke frames. Since the model checking algorithm is defined on the Kripke model, it makes sense to simplify the encoding of the polyhedral model. First, we present the definition of simplicial bisimulation for polyhedral models, along with two key results concerning logical equivalence. We then present the definition of simplicial bisimulation for Kripke frames with a reachability operator and prove the Hennessy-Milner result for the language under consideration. The second subsection consists only of original definitions and results that have not yet been considered.

NB: in the current section, we work with the logical equivalence for language $\mathcal{L}_{\Box\gamma}$, and instead of $\equiv_{\mathcal{L}_{\Box\gamma}}$ we write $\equiv$.

2.4.1 Simplicial Bisimulation on Polyhedra

To accommodate the addition of the reachability operator in the logical language, the definition of bisimilarity incorporates the point-wise lifting of a relation to a path, which is formally defined below. Throughout the following discussion, let us consider a fixed polyhedral model $\mathcal{E} = (P, K, V)$.

Definition 2.4.1. Given a relation $R \subseteq P \times P$, let the R *extension* to paths be a binary relation between paths $\hat{R}$ such that $\pi_1 \hat{R} \pi_2$ if for all $t \in [0,1]$ we have $\pi_1(t) R \pi_2(t)$.

Definition 2.4.2. A binary relation $\sim \subseteq P \times P$ is a *simplicial bisimulation* if for all x, y with $x \sim y$:

1. for all $p \in \text{Prop}, x \in V(p) \iff y \in V(p)$;
2. for each simplicial path π_x such that $\pi_x(0) = x$, 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 such that $\pi_y(0) = y$, 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*.

Theorem 2.4.3. Logical equivalence is a simplicial bisimulation.

Proof. See Theorem 6.3 in [Bez+22]. □

Theorem 2.4.4. Each simplicial bisimulation is included in logical equivalence.

Proof. See [Bez+22, Theorem 6.4]. □

Corollary 2.4.5. In a polyhedral model, the largest simplicial bisimulation always exists and coincides with logical equivalence.

2.4.2 Simplicial Bisimulation on Kripke frames

We will now delve into the study of simplicial bisimulation on Kripke frames and address the notion of path equivalence, which must be analogous to the definition of 2.4.1. The challenge for Kripke models lies in the presence of a notion of the length of a path. While in topological spaces we have no notion of the length of a path, and any path can be 'stretched' as far as one needs to, each path in a Kripke frame is a map from a finite subset of $\mathbb{N}$. Consequently, it is difficult to determine equivalent paths of different lengths. This problem can be solved by introducing the concept of "path extension".

Definition 2.4.6. Suppose $\pi : \{0, \ldots, k\} \to W$ is a $\pm$-path on a Kripke model $\mathcal{M} = (W, \sqsubseteq, V)$. If $r_0, \ldots, r_k \in \mathbb{N}$ and $n = \sum_{i=0}^{k} r_i$, then an *extension of a path* π is a path $\pi' : \{0, \ldots, n\} \to W$ s.t.:

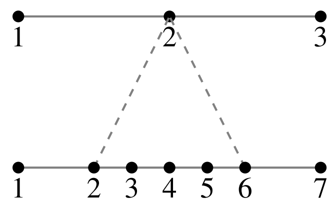

Figure 2.4: Example of the extension of a path.

1. $rng(\pi) = rng(\pi')$;
2. $\pi'(\{\mathfrak{k}_{i-1}+1, \ldots, \mathfrak{k}_i\}) = \pi(i)$, for $i > 0$
   and $\pi'(\{0, \ldots, \mathfrak{k}_0\}) = \pi(i)$, for $i = 0$.
   where $\mathfrak{k}i = \sum\limits_{j=0}^{i} r_j$.

Clearly, even the shortest path $\pi : \{0, 1, 2\} \to W$ has infinitely many extensions. For example $\pi' : \{0, 1, 2, 3, 4, 5, 6, 7\} \to W$ where $\pi(1) = \pi'(\{1, 2, 3, 4, 5, 6\})$ and thus $r_0 = 0, r_1 = 6, r_2 = 0$. To get the intuition, see the figure 2.4. Since our Kripke frames are reflexive, we have that an extension can be obtained by making finitely many "jumps" on each point.

Now we can define what it means for paths to be equivalent i.e., definition analogous to 2.4.6.

Definition 2.4.7. Let R be a relation on W. We call $\dot{R}$ an *extension of R to paths* if $\pi_1 \dot{R} \pi_2$ iff :

- there is an extension π' of π_1, s.t. for all $t \in dom(\pi_2) : \pi'(t) R \pi_2(t)$;

or

- there is an extension π' of π_2, s.t. for all $t \in dom(\pi_1) : \pi_1(t) R \pi'(t)$.

Observe that this definition covers two cases: when path π_1 is shorter than π_2 and vice versa, namely when π_2 is shorter than π_1.

Now we can define simplicial bisimulation on Kripke frames.

Definition 2.4.8. Suppose $\mathcal{M} = (W, \sqsubseteq, V)$ is a Kripke model.

A relation $\sim \subseteq W \times W$ is a *Kripke simplicial bisimulation* if for all $x, y \in W$ such that $x \sim y$:

1. $x \in V(p)$ iff $y \in V(p)$ for all $p \in Var$;

2. if $\pi_x : \{0, \ldots, k\} \to W$ is a $\pm$ - path, s.t. $\pi_x(0) = x$, then there exists $\pm$ - path $\pi_y : \{0, \ldots, k'\} \to W$ s.t. $\pi_y(0) = y$ and $\pi_x \dot{\sim} \pi_y$.

3. if $\pi_y : \{0, \ldots, k\} \to W$ is a $\pm$ - path, s.t. $\pi_y(0) = y$, then there exists $\pm$ - path $\pi_x : \{0, \ldots, k'\} \to W$ s.t. $\pi_x(0) = x$ and $\pi_x \dot{\sim} \pi_y$.

If $x \sim y$, we shall call such points *bisimilar*.

Since the Kripke models we are interested in are always finite due to the finiteness of each triangulation, we will only consider finite Kripke models. This restriction allows us to connect simplicial bisimulation on Kripke frames very neatly to logical equivalence. First, we prove that bisimulation between two points implies logical equivalence.

Theorem 2.4.9. If two points in a finite Kripke model are simplicially bisimilar, they are logically equivalent.

Proof. Assume that $x \sim y$. We have to show that $x \equiv y$.

We will prove this statement by induction on the complexity of formula φ. The case for propositional letters follows from the definition 2.4.8. Cases of Boolean connectives are trivial. Since modality $\Box$ can be expressed using γ, we will consider only the case for γ.

We have to show that $\mathcal{M}, x \vDash \gamma(\varphi, \psi)$ if and only if $\mathcal{M}, y \vDash \gamma(\varphi, \psi)$. The proofs of both directions are analogous, so we will prove only the $\Rightarrow$ direction.

Assume that $\mathcal{M}, x \vDash \gamma(\varphi, \psi)$. Then there is a path $\pi_x : \{0, \ldots, k\} \to W$, such that $\pi_x(x) = 0$, $\pi_x(\{1, \ldots, k-1\}) \subseteq [\![\varphi]\!]$ and $\pi_x(k) \in [\![\psi]\!]$. Since $x \sim y$, we have that there is a path $\pi_y : \{0, \ldots, k'\}$, such that $\pi_y(0) = y$ and $\pi_x \dot{\sim} \pi_y$. Therefore, there are two cases:

- there is an extension π' of π_x, s.t. for all $t : \pi'(t) \sim \pi_y(t)$;

or

- there is an extension π' of π_y, s.t. for all $t : \pi_x(t) \sim \pi'(t)$;

Let us consider the first case. We obtain that for every $t \in dom(\pi')$ there exists $i \in \{1, \ldots, k\}$:

- either $i < k$ and $\pi'(t) = \pi(i)$ and therefore by inductive hypothesis we have $\mathcal{M}, \pi_y(t) \vDash \varphi$;
- or $i = k$ and $\pi'(t) = \pi(k)$ and again by inductive hypothesis: $\mathcal{M}, \pi_y(t) \vDash \psi$.

Hence, we have that $\pi_y(\{1, \ldots, k'-1\}) \subseteq [\![\varphi]\!]$ and $\pi_y(k') \in [\![\psi]\!]$. Therefore, $\mathcal{M}, y \vDash \gamma(\varphi, \psi)$.

In the second case, when there is an extension π' of π_y, such that for all t it holds that $\pi_x(t) \sim \pi'(t)$, then we obtain directly by the inductive hypothesis that $\pi_x(\{1, \ldots, k-1\}) \subseteq [\![\varphi]\!]$ and $\pi_x(k) \in [\![\psi]\!]$. We conclude that $\mathcal{M}, y \vDash \gamma(\varphi, \psi)$. □

This is the least condition that we always require from bisimulation. Having it, we justify our definition of bisimilarity. Now we will prove a slightly more complicated theorem: logical equivalence is bisimulation. This can be seen as the main original result of the chapter, which draws a line below the bisimulation research initiated in [Bez+22].

Theorem 2.4.10. Logical equivalence on a finite Kripke model is Kripke simplicial bisimulation.

Proof. Suppose that $x \equiv y$. Then clearly, the first item of definition 2.4.8 holds. Let us show that the second item holds as well. We omit the proof of the third item since it is analogous to the second one.

Suppose that $\pi_x : \{0, \ldots, k\} \to W$ is a $\pm$ - path, s.t. $\pi_x(0) = x$. We will prove by induction on k that there exists a path $\pm$ - path $\pi_y : \{0, \ldots, k'\} \to \widetilde{K}$ s.t. $\pi_y(0) = y$ and $\pi_x \dot{\equiv} \pi_y$.

Base: k=2

If $\pi_x : \{0, 1, 2\} \to \widetilde{K}$, take classes of equivalence C_1 and C_2 s.t. $\pi_x(1) \in C_1$ and $\pi_x(2) \in C_2$. Since our model is finite, every class of equivalence C_i is characterized by some formula φ_i. Take φ_1, φ_2 that characterize C_1 and C_2 respectively. Then $\mathcal{M}, x \vDash \gamma(\varphi_1, \varphi_2)$. By assumption we have that $\mathcal{M}, y \vDash \gamma(\varphi_1, \varphi_2)$, i.e. there is a $\pm$ - path $\pi_y : \{0, \ldots, k\} \to W$ s.t. $\pi_y(0) = y$, $\pi_y(\{1, \ldots, k-1\}) \subseteq [\![\varphi_1]\!]$ and $\pi_y(k) \in [\![\varphi_2]\!]$. Now we have to check, whether $\pi_x \dot{\equiv} \pi_y$. Let us take the extension π'_x of π_x s.t. $\pi'_x : \{0, \ldots, k\} \to W$ and:

$$\pi'_x(t) = \begin{cases} x, & \text{if } t = 0 \\ \pi_x(1), & 1 \leq t < k \\ \pi_x(2), & t = k \end{cases}$$

By the definition of this path, we have that $\pi'_x(\{1, \ldots, k-1\}) \subseteq [\![\varphi_1]\!]$ and $\pi'_x(k) \in [\![\varphi_2]\!]$. Therefore, for every $t \in \{0, \ldots, k\}$: $\pi'_x(t)$ is in the same equivalence class as $\pi_y(t)$.

Step $\boldsymbol{k \curvearrowright k+1}$:

Suppose that for every path π_x of length k from x we can find a path π_y from y, such that $\pi_x \dot{\equiv} \pi_y$. Now let us show that this also holds for every path of length $k+1$.

Suppose $\pi_x : \{0, \ldots, k, k+1\} \to W$ is a path s.t. $\pi_x(0) = x$. Let us take path $\pi'_x : \{0, \ldots, k\} \to W$ s.t. $\pi'_x = \pi_x \upharpoonright \{0, \ldots, k\}$. By induction hypothesis we have that there is a path $\pi'_y : \{0, \ldots, m\} \to W$ s.t. $\pi'_y(0) = y$ and $\pi'_x \dot{\equiv} \pi'_y$, i.e. there is an extension π'' of π'_x s.t. $\forall t : \pi''(t) \equiv \pi'_y(t)$, or π'' is an extension of π'_y and $\forall t : \pi'_x(t) \equiv \pi''(t)$.

Let us consider the first case when there is an extension $\pi'' : \{0, \ldots, m\} \to W$ of π'_x, s.t. $\forall t \in dom(\pi'_y) : \pi''(t) \equiv \pi'_y(t)$. By the definition of extension there are $r_0, \ldots, r_k$, s.t. $\sum\limits_{i=0}^{k} r_i = m$ with the conditions from definition 2.4.6. Then we have that $\pi''(m) \equiv \pi'_y(m)$ and at the same time $\pi''(m) = \pi(k)$. Take a path $\hat{\pi}_x : \{0, 1, 2\} \to W$, s.t.:

$$\dot{\pi}_x(z) = \begin{cases} \pi(k), & \text{if } z = 0, 1; \\ \pi(k+1), & \text{if } z = 2. \end{cases}$$

Use the reasoning from the base case on points $\pi(k)$, $\pi'_y(m)$ and path $\hat{\pi}_x$. We obtain that there is a path $\hat{\pi}_y : \{0, \ldots, t\} \to W$ s.t. $\dot{\pi}_x \dot{\equiv} \hat{\pi}_y$. Then by definition there exists an extension $\pi^* : \{0, \ldots, t\} \to W$ of $\dot{\pi}_x$ with r'_0, r'_1, r'_2, s.t. $r'_0 + r'_1 + r'_2 = t$, again with the conditions from definition 2.4.6.

Thus, let us take $\pi_y : \{0, \ldots, m+t+1\} \to W$ defined in the following way:

$$\pi_y(z) = \begin{cases} \pi'_y(z), & \text{if } z \in \{0, \ldots, m\}; \\ \dot{\pi}_y(z - m - 1), & \text{if } z \in \{m+1, \ldots, m+t+1\}. \end{cases}$$

Now let us take $r^*_0 = r_0, \ldots, r^*_{k-1} = r_{k-1}, r^*_k = r_k + r'_0 + r'_1, r^*_{k+1} = r'_2$ and define π^*_x:

- $\pi^*_x(\{\mathfrak{k}^*_{i-1}, \ldots, \mathfrak{k}^*_i\}) = \pi_x(i), \text{ for } i > 0;$
- $\pi^*_x(\{0, \ldots, \mathfrak{k}^*_0\}) = \pi_x(0), \text{ for } i = 0,$

  where $\mathfrak{k}^*_j = \sum\limits_{i=0}^{j} r^*_i$.

Then π^*_x is an extension of π_x. Note that the length of π_y is the same as the length of π^*_x, namely $m + t + 1$, and by construction $\forall t : \pi_y(t) \equiv \pi^*_x(t)$. We conclude that $\pi_x \dot{\equiv} \pi_y$.

Assume that we are in the second case, when: $\pi'' : \{0, \ldots, k\} \to W$ is an extension of π'_y, s.t. $\forall t \in dom(\pi'_x) : \pi'_x(t) \equiv \pi''(t)$. First, fix again $r_0, \ldots, r_k$, s.t. $\sum_{i=0}^{k} r_i = k$ with the properties from definition 2.4.6.

Now define again a path $\dot{\pi}_x(z)$, s.t.:

$$\dot{\pi}_x(z) = \begin{cases} \pi(k), & \text{if } z = 0, 1; \\ \pi(k+1), & \text{if } z = 2. \end{cases}$$

and use the base case reasoning on points $\pi(k)$, $\pi''(k)$ and path $\dot{\pi}_x$. We obtain:

- Path $\dot{\pi}_y : \{0, \ldots, t\} \to W$
- Extension $\pi^* : \{0, \ldots, t\} \to W$ of $\dot{\pi}_x$;
- $r'_0, r'_1, r'_2 \in \mathbb{N}$, s.t. $r'_0 + r'_1 + r'_2 = t$ and conditions from definition 2.4.6.

We can define now $\pi_y : \{0, \ldots, k+t+1\} \to W$ in the following way:

$$\pi_y(z) = \begin{cases} \pi''(z), & \text{if } z \in \{0, \ldots, k\}; \\ \dot{\pi}_y(z-k-1), & \text{if } z \in \{k+1, \ldots, t+k+1\}. \end{cases}$$

Then, let us take $r'_1, r'_2, r'_3 \in \mathbb{N}$ and extension π^*_x of π_x, s.t. $\pi^*_x : \{0, \ldots, k+t+1\} \to W$, where:

$$\pi^*_x(z) = \begin{cases} \pi_x(z), & \text{if } z \in \{0, \ldots, k\} \\ \pi^*(z-k-1), & \text{if } z \in \{k+1, \ldots, t+k+1\}. \end{cases}$$

Note that the length of π_y is the same as the length of π^*_x, namely $k+t+1$. Thus, by construction: $\forall t \in dom(\pi_y) : \pi^*_x(t) \equiv \pi_y(t)$. We conclude that $\pi_x \dot{\equiv} \pi_y$. $\square$

We conclude this chapter with the following corollary:

Corollary 2.4.11. In a finite Kripke model, the largest simplicial bisimulation always exists, coinciding with logical equivalence.

Proof. Straightforward from Theorems 2.4.9 and 2.4.10. $\square$

We conclude this section by summarizing the result of the chapter.

- We defined the central geometric notions for this work: polyhedron, triangulation, and simplexes.
- We introduced the language $\mathcal{L}_{\Box\gamma}$ and provided the semantics for this language on polyhedral models.
- We defined the notion of a Kripke frame and explained how to construct a Kripke encoding for a polyhedral model.
- We proved that every formula that is true in a point x in a polyhedral model $\mathcal{E}$ if and only if it is true in $\widetilde{\sigma}^x$ in the encoding $\mathcal{M}(\mathcal{E})$ of $\mathcal{E}$.
- We introduced a self-developed definition of simplicial bisimulation on finite Kripke frames for formulas in the language $\mathcal{L}_{\Box\gamma}$ and proved for the first time the Hennessy-Milner result for it.

Chapter 3

Modelling polyhedral dynamics

In this chapter, we extend our framework to the setting of dynamic systems and dynamic frames. This direction of research lies in the domain of spatial-temporal logic. This topic has been widely studied before, especially from the side of topological spaces. In [ADN97], the authors extend the basic modal logic language with a temporal modal operator "next" and obtain interesting completeness results. This research direction was taken further by Kremer and Mints [KM07], where the "henceforth" operator was added to the logic. A broad perspective on this field was also presented in [Kon+07]. In our case, we extend language $\mathcal{L}_{\Box,\gamma}$ with modality "next" while also changing the underlying structures from topological spaces to polyhedra. Following the strategy outlined in sections 1-3 of the previous chapter, we will show that it is possible to reduce the question of the truth of a formula in our dynamic model to the question of the truth of a formula on a dynamic finite Kripke model, which we will call its encoding.

3.1 Dynamic Polyhedral semantics

In previous works on dynamic semantics, such as [ADN97] and [KM07], authors used a functional relation R to model the dynamics. Thus, if a proposition p is true at a point $R(x)$, p will be true at x at the next moment. In other words, this means that $x \in [\![\langle \mathrm{R} \rangle p]\!]$ iff $R(x) \in [\![p]\!]$ iff $x \in R^{-1}([\![p]\!])$. Thus, we arrive at the following equation:

$$[\![\langle \mathrm{R} \rangle p]\!]^P = R^{-1}([\![p]\!]^P)$$

However, we want to keep our dynamics definition as general as possible. That is why we will waive the functionality requirement of relation R.

Definition 3.1.1. Let P be a polyhedron with triangulation K, and R be a relation on P. Then:

1. $\mathcal{D} = (P, K, R)$ is a *dynamic system*;

2. $\mathcal{O} = (P, K, R, V)$ is a *dynamic model*, whenever (P, K, V) is a polyhedral model.

For ease of notation, we will sometimes write $(\mathcal{D}, V)$ instead of writing in full (P, K, R, V).

We will now define a new modal language and its models.

Definition 3.1.2. The modal language $\mathcal{L}_{\gamma,\langle\mathrm{R}\rangle}$ is defined as the language $\mathcal{L}_{\Box\gamma}$ extended with a unary modal operator $\langle\mathrm{R}\rangle$. The well-formed formulas of the language $\mathcal{L}_{\gamma,\langle\mathrm{R}\rangle}$ are defined by the rule:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box\varphi \mid \gamma(\varphi, \varphi) \mid \langle\mathrm{R}\rangle\varphi$$

where p is an atomic proposition from Prop.

The new modality $\langle\mathrm{R}\rangle$ is analogous to $\Diamond$, and we define $[\mathrm{R}]\varphi := \neg\langle\mathrm{R}\rangle\neg\varphi$. Indeed, our operator $\langle\mathrm{R}\rangle$ models the relation R in the definition of a dynamic system.

Definition 3.1.3. Suppose $\mathcal{O} = (P, K, R, V)$ is a dynamic model. We define recursively when a formula φ in the language $\mathcal{L}_{\gamma,\langle\mathrm{R}\rangle}$ is satisfied at a point x (notation: $\mathcal{O}, x \vDash \varphi$):

$$\begin{array}{lcl}
\mathcal{O}, x \vDash p & \iff & x \in V(p) \text{ for } p \in \mathrm{Prop} \\
\mathcal{O}, x \vDash \neg\varphi & \iff & \mathcal{O}, x \nvDash \varphi \\
\mathcal{O}, x \vDash \varphi \wedge \psi & \iff & \mathcal{O}, x \vDash \varphi \text{ and } \mathcal{O}, x \vDash \psi \\
\mathcal{O}, x \vDash \Box\varphi & \iff & x \in \mathrm{Int}\left(\llbracket\varphi\rrbracket^{\mathcal{O}}\right) \\
\mathcal{O}, x \vDash \gamma(\varphi, \psi) & \iff & \text{there exists a path } \pi \text{ such that} \\
 & & \pi(0) = x, \pi(1) \in \llbracket\psi\rrbracket^{\mathcal{O}} \text{ and } \pi((0,1)) \subseteq \llbracket\varphi\rrbracket^{\mathcal{O}} \\
\mathcal{O}, x \vDash \langle\mathrm{R}\rangle\varphi & \iff & x \in f^{-1}(\llbracket\varphi\rrbracket^{\mathcal{D}}).
\end{array}$$

As before, $\llbracket\varphi\rrbracket^{\mathcal{O}}$ is the set of all points of P where φ is true.

We also define global truth in our models.

Definition 3.1.4. We will say that φ is *globally true* in a dynamic model $\mathcal{O} = (P, K, R, V)$ (notation: $\mathcal{O} \vDash \varphi$) if for every point $x \in P : \mathcal{O}, x \vDash \varphi$.

In the previous chapter, the main result is Theorem 2.3.15, which relates the truth of formulas on a polyhedral model $\mathcal{E}$ and with the truth of formulas on its encoding $\mathcal{M}(\mathcal{E})$. However, this result would not be possible without showing that $[\![\varphi]\!]^{\mathcal{M}(\mathcal{E})}$ is the union of relative interiors of simplexes. That is why our primary goal for this section is to prove the same result for our dynamic models. Since the only difference between dynamic models and polyhedral models is the additional relation R, we have to show that $[\![\langle \mathrm{R} \rangle \varphi]\!]^{\mathcal{O}}$ is the union of relative interiors of simplexes. It is clear that not every relation R will have this property. Then we have to restrict R to obtain the desired result. Following the tradition of defining dynamic relations as a function, we begin our investigation with the notion of *polyhedral function*.

Definition 3.1.5. Let P be a polyhedron and K its triangulation. Define $Sub_K(P) \subseteq Sub(P)$ as:

$$P' \in Sub_K(\mathrm{P}) \text{ iff } \mathrm{P}' = |K'| \text{ for some } K' \text{ subcomplex of } K.$$

Definition 3.1.6. Let P be a polyhedron and K its triangulation. We call $f : P \to P$ a *polyhedral map*, whenever f is continuous and $f^{-1}(P') \in Sub_K(P)$ for every $P' \in Sub_K(P)$.

We have a useful technical lemma.

Lemma 3.1.7. Let P be a polyhedron and K its triangulation. Then $P' \in Sub_K(P)$ iff there is a downset $C \subseteq \widetilde{K}$ such that $P' = \bigcup C$.

Proof. The right-to-left direction is trivial. Let us prove left to right. Assume $P' \in Sub_K(P)$. Then $P' = |K'|$ for some $K' \subseteq K$. Take $D = \{\widetilde{\tau} \mid \exists \sigma \in K' : \widetilde{\sigma} \precsim \widetilde{\tau}\}$. Then D is downclosed and $\bigcup D = |K'|$. □

To have a result analogous to Proposition 2.2.7, we need somewhat stronger: that a preimage of an element of $\mathfrak{U}(K)$ is in $\mathfrak{U}(K)$. The following lemma establishes this property.

Lemma 3.1.8. Let P be a polyhedron, K its triangulation, and $f : P \to P$ a polyhedral map. Then for every $\widetilde{\sigma} \in \widetilde{K}$ yields $f^{-1}(\widetilde{\sigma}) = \bigcup\{\widetilde{\delta} \mid f(\widetilde{\delta}) \subseteq \widetilde{\sigma}\}$.

Proof. We proceed by induction on levels of $\widetilde{K}$.

Case: 0-level.

Take $\widetilde{\sigma} \in \widetilde{K}_0$. Then $\widetilde{\sigma}$ is a 0 -simplex and $\widetilde{\sigma} = \sigma$. Since f is polyhedral, $f^{-1}(\widetilde{\sigma}) = |K'|$ such that $f^{-1}(\widetilde{\sigma}) \in Sub_K(|K|)$. By Lemma 3.1.7, it follows that there is a downset $C \subseteq \widetilde{K}$ such that $\bigcup C = |K'|$. Thus, $f^{-1}(\widetilde{\sigma}) = \bigcup C$, and therefore for every $\widetilde{\delta} \in C$ we have $f(\widetilde{\delta}) \subseteq \widetilde{\sigma}$.

Case: i-level.

Suppose that for all levels $l < i$ holds that if $\widetilde{\theta} \in \widetilde{K}_l$, then $f^{-1}(\widetilde{\theta}) = \bigcup\{\widetilde{\delta} \mid f(\widetilde{\delta}) \subseteq \widetilde{\sigma}\}$. Take $\widetilde{\sigma} \in \widetilde{K}_i$. Then $f^{-1}(\sigma) = |K'|$ such that $f^{-1}(\sigma) \in Sub_K(P)$. At the same time we have that $\sigma = \widetilde{\sigma} \cup \bigcup D$, where $D = \{\widetilde{\tau} \mid \widetilde{\tau} \precsim \widetilde{\sigma}, \widetilde{\tau} \neq \widetilde{\sigma}\}$. Then clearly, if $\widetilde{\tau} \in D$, then $\tau \in K_l$ for some $l < i$. So, we have that $f^{-1}(\sigma) = f^{-1}(\widetilde{\sigma} \cup \bigcup D) = f^{-1}(\widetilde{\sigma}) \cup \bigcup f^{-1}(D)$[1]. Thus, $f^{-1}(\widetilde{\sigma}) = |K'| \setminus \bigcup f^{-1}(D)$. To show the desired equality, observe that $\bigcup\{\widetilde{\delta} \mid f(\widetilde{\delta}) \subseteq \widetilde{\sigma}\} \subseteq f^{-1}(\widetilde{\sigma})$.

To show the other inclusion, take $x \in f^{-1}(\widetilde{\sigma})$. Then $x \in |K'|$ and $x \notin \bigcup f^{-1}(D)$. Take $\widetilde{\sigma}^x$. We will show that $\widetilde{\sigma}^x \subseteq f^{-1}(\widetilde{\sigma})$. First, let us check that $\widetilde{\sigma}^x \subseteq |K'|$. Since $x \in |K'|$, then there is a simplex $\tau \in K'$ such that $x \in \widetilde{\sigma}^x \cap \tau$. By Lemma 2.3.12 it yields $\sigma^x \preccurlyeq \tau$, and therefore $\widetilde{\sigma}^x \subseteq |K'|$. Assume that $\widetilde{\sigma}^x \cap \bigcup f^{-1}(D) \neq \varnothing$. Then there is $\widetilde{\delta} \in D$ such that $\widetilde{\sigma}^x \cap f^{-1}(\widetilde{\delta}) \neq \varnothing$. By the condition on elements in D we obtain that $f^{-1}(\delta) = \bigcup\{\widetilde{\theta} \mid f(\widetilde{\theta}) \subseteq \widetilde{\delta}\}$. Hence, $\widetilde{\sigma}^x \in \{\widetilde{\theta} \mid f(\widetilde{\theta}) \subseteq \widetilde{\delta}\}$. But then we have that $x \in \bigcup f^{-1}(D)$, which contradicts our assumption. We conclude that $\widetilde{\sigma}^x \subseteq f^{-1}(\widetilde{\sigma})$, and therefore $x \in \bigcup\{\widetilde{\delta} \mid f(\widetilde{\delta}) \subseteq \widetilde{\sigma}\}$. $\square$

Let us now introduce an adjusted definition of dynamic systems and models, and instead of taking any relation R, take a polyhedral function.

Definition 3.1.9.

1. Let $\mathcal{D} = (P, K, R)$ be a dynamic system. If R is a polyhedral function, we will call $\mathcal{D}$ a *dynamic polyhedral structure*.
2. If $\mathcal{O} = (\mathcal{D}, V)$ is a dynamic model, with $\mathcal{D}$ being a dynamic polyhedral structure, then we will call $\mathcal{O}$ a *dynamic polyhedral model*.

From now on, we will continue working with dynamic polyhedral models and systems, specifying particular cases when R is not polyhedral. For the simplicity of notation, we will write f instead of R when R is functional. So, now our goal is to prove that an analogue of Proposition 2.2.7 holds for dynamic polyhedral models and formulas in language $\mathcal{L}_{\gamma\langle\mathrm{R}\rangle}$.

Lemma 3.1.10. Let $\mathcal{O} = (P, K, f, V)$ be a dynamic polyhedral model. For each cell $\widetilde{\sigma} \in \widetilde{K}$, for all points $x, y \in \widetilde{\sigma}$, it is true that $x \equiv_{\mathcal{L}_{\gamma,\langle\mathrm{R}\rangle}} y$.

Proof. The proof is by induction on the complexity of a formula. The case of propositional letters follows by definition. The cases of boolean operators are trivial. The cases of $\Box\varphi$ and $\gamma(\varpi, \psi)$ is the same as for formulas $\Box\varphi$ and $\gamma(\varpi, \psi)$ in Lemma 2.2.6, that is why we prove only the case for $\langle\mathrm{R}\rangle\varphi$.

[1] We slightly abuse the notation here: $f^{-1}(A) = \bigcup\{f^{-1}(x) \mid x \in A\}$.

Take an element $\widetilde{\sigma} \in \widetilde{K}$ and $x, y \in \widetilde{\sigma}$. Suppose that $\mathcal{E}, x \vDash \langle \mathrm{R} \rangle \varphi$. Then $x \in f^{-1}(\llbracket \varphi \rrbracket)$. Take $f(x) \in \llbracket \varphi \rrbracket$. Then $f(x) \in \widetilde{\tau}$ for some $\widetilde{\tau} \in \widetilde{K}$. Let us show that $f(y) \in \widetilde{\tau}$, then by inductive hypothesis we will have that $y \in f^{-1}(\llbracket \varphi \rrbracket)$. By Lemma 3.1.8 yields $f^{-1}(\widetilde{\tau}) = \bigcup \{ \widetilde{\delta} \mid f(\widetilde{\delta}) \subseteq \widetilde{\tau} \}$. Therefore, $x \in \bigcup \{ \widetilde{\delta} \mid f(\widetilde{\delta}) \subseteq \widetilde{\tau} \}$. Thus, $x \in \widetilde{\delta}$ for some $f(\widetilde{\delta}) \subseteq \widetilde{\tau}$. Since $\widetilde{K}$ is a set of pairwise-disjoint elements, it follows that $\widetilde{\delta} = \widetilde{\sigma}$. Hence, $f(y) \in \llbracket \varphi \rrbracket$. □

So, the desired result follows.

Proposition 3.1.11. Let $\mathcal{O} = (P, K, f, V)$ be a dynamic polyhedral model, and φ a formula in $\mathcal{L}_{\gamma\Box}$. Then $\llbracket \varphi \rrbracket^{\mathcal{O}} \in \mathfrak{U}(K)$.

Proof. Similar to the proof of Theorem 2.2.7. □

3.2 Dynamic Kripke frames

The following section defines dynamic Kripke frames and establishes the correspondence between them and dynamic polyhedral systems. Our basic definition differs slightly from the definition of dynamic Kripke frames in classical papers on dynamic logics such as [KM07]. The main difference is that we do not require additional properties on our relation R. However, as we will see, the dynamic Kripke frame from [KM07] will be a special case of our dynamic frames. In the second half of this section, we will explain how to construct an encoding of a dynamic polyhedral model and then conclude this section with a theorem analogous to the theorem 2.3.15.

As in the previous section, we first give a very general definition of dynamic Kripke frames.

Definition 3.2.1.

1. A *dynamic Kripke frame* is a tuple $\mathcal{F}^d = (W, \sqsubseteq, R)$, where $(W, \sqsubseteq)$ is a Kripke frame and R is a binary relation on W.

2. A *dynamic Kripke model* is a tuple $\mathcal{M}^d = (W, \sqsubseteq, R, V)$ such that $(W, \sqsubseteq, R)$ is a dynamic Kripke frame and $(W, \sqsubseteq, V)$ is a Kripke model.

For the ease of notation, we will sometimes write $(\mathcal{F}^d, V)$ instead of $(W, \sqsubseteq, R, V)$.

The semantics of formulas in $\mathcal{L}_{\gamma,\langle \mathrm{R} \rangle}$ is defined as follows:

Definition 3.2.2. Let $\mathcal{M}^d = (W, \sqsubseteq, R, V)$ be a dynamic Kripke model and $x \in W$. We define recursively when a formula φ in language $\mathcal{L}_{\gamma,\langle\mathrm{R}\rangle}$ is satisfied at a point x (notation: $\mathcal{M}^d, x \vDash \varphi$):

$$
\begin{array}{lcl}
\mathcal{M}^d, x \vDash p & \iff & x \in V(p) \text{ for } p \in \mathrm{Prop} \\
\mathcal{M}^d, x \vDash \neg\varphi & \iff & \mathcal{M}^d, x \nvDash \varphi \\
\mathcal{M}^d, x \vDash \varphi \wedge \psi & \iff & \mathcal{M}^d, x \vDash \varphi \text{ and } \mathcal{M}^d, x \vDash \psi \\
\mathcal{M}^d, x \vDash \Box\varphi & \iff & \forall y : x \sqsubseteq y \Rightarrow \mathcal{M}^d, y \vDash \varphi \\
\mathcal{M}^d, x \vDash \gamma(\varphi, \psi) & \iff & \text{there exists a } \pm\text{- path } \pi : \{0, \ldots, k\} \to W \text{ such that} \\
 & & \pi(0) = x, \pi(k) \in [\![\psi]\!]^{\mathcal{M}^d} \text{ and } \pi(\{1, \ldots, k-1\}) \subseteq [\![\varphi]\!]^{\mathcal{M}^d} \\
\mathcal{M}^d, x \vDash \langle\mathrm{R}\rangle\varphi & \iff & x \in R^{-1}([\![\varphi]\!]^{\mathcal{M}^d}).
\end{array}
$$

As before, $[\![\varphi]\!]^{\mathcal{M}^d}$ is the set of all points in $\mathcal{M}^d$, where φ is true.

Definition 3.2.3. Let $\mathcal{M}^d = (W, \sqsubseteq, R, V)$ be a dynamic Kripke model. We will say that φ is *globally true* on $\mathcal{M}^d$ (notation: $\mathcal{M}^d \vDash \varphi$), if for every point $x \in W$: $\mathcal{M}^d, x \vDash \varphi$.

We must impose some restrictions on the relations we consider to find a correspondence between dynamic polyhedral structures and dynamic Kripke frames. Let us consider the class of *monotone functions*.

Definition 3.2.4. Let $\mathcal{F} = (W, \sqsubseteq)$ be a Kripke frame, and $f : W \to W$ be a function. Then we call f a *monotone function* on $\mathcal{F}$, if $x \sqsubseteq y$ implies $f(x) \sqsubseteq f(y)$.

Definition 3.2.5. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic Kripke frame, then:

1. We call $\mathcal{F}^d$ a *monotone dynamic frame* if R is a monotone function, .
2. We call $\mathcal{M}^d = (\mathcal{F}^d, V)$ a *monotone dynamic model* if $\mathcal{F}^d$ is a monotone dynamic frame.

The following lemma provides a characterization of monotone functions in terms of downsets.

Lemma 3.2.6. Suppose $f : W \to W$ is a function. Then f is monotone iff for every $C \subseteq W$: if C is a downset, then $f^{-1}(C)$ is a downset.

Proof. ($\Rightarrow$) Assume that $y \in f^{-1}(C)$ for some downset set C, and $x \sqsubseteq y$. Then we have that $f(y) \in C$. By monotonicity of f: $f(x) \sqsupseteq f(y)$. Thus, $f(x) \in C$ and $x \in f^{-1}(C)$.

($\Leftarrow$) Assume that $x \sqsubseteq y$. To show that $f(x) \sqsubseteq f(y)$, we will show that $f(x) \in\downarrow f(y)$. Take $f^{-1}(\downarrow f(y))$. Since $\downarrow f(y)$ is downset, $f^{-1}(\downarrow f(y))$ is also downset. Since $y \in f^{-1}(\downarrow f(y))$, we have that $x \in f^{-1}(\downarrow f(y))$. Thus, $f(x) \in\downarrow f(y)$. $\square$

The next challenge is to define the corresponding dynamic Kripke frame using the information stored in the dynamic polyhedral system (P, K, f). The domain of this Kripke frame should be $\widetilde{K}$, and the poset relation has to be defined as $\precsim$. The only question is how to define the monotone function on $\widetilde{K}$, using the information from polyhedral function f. Using Lemma 3.1.8 we can define $f_K : \widetilde{K} \to \widetilde{K}$ as follows:

Definition 3.2.7. Let (P, K, f) be a dynamic polyhedral system. Define $f_K : \widetilde{K} \to \widetilde{K}$ in the following way:

$$f_K(\widetilde{\sigma}) = \widetilde{\tau} \text{ iff } f(\widetilde{\sigma}) \subseteq \widetilde{\tau}.$$

Observe that with this definition, we have the following property for every $A \subseteq \widetilde{K}$:

$$\bigcup (f_K)^{-1}(A) = f^{-1}(\bigcup A).$$

To define a dynamic Kripke frame, we must show that f_K is monotone.

Lemma 3.2.8. Let $\mathcal{D} = (P, K, f)$ be a dynamic polyhedral system. Then function $f_K : \widetilde{K} \to \widetilde{K}$ is monotone.

Proof. We will show that $(f_K)^{-1}(C)$ is a downset for every downset C, and then use Lemma 3.2.6.

Take a downset set C in $\widetilde{K}$. Then $\bigcup C$ is a polyhedron by Lemma 2.3.13. Thus, $f^{-1}(\bigcup C)$ is also a polyhedron $P' \in Sub_K(P)$ and therefore by the same Lemma 2.3.13 corresponds to a downset subset $\widetilde{K}'$ of $\widetilde{K}$. Since $f^{-1}(\bigcup C) = \bigcup (f_K)^{-1}(C)$ and both $(f_K)^{-1}(C)$ and C' are subsets of $\widetilde{K}$, we conclude that $(f_K)^{-1}(C) = C'$. $\square$

Definition 3.2.9. Let $\mathcal{D} = (P, K, f)$ be a dynamic polyhedral system. We define its *encoding* $\mathcal{F}(\mathcal{D}) = (\widetilde{K}, \precsim, f_K)$, as follows:

1. $(\widetilde{K}, \precsim)$ is the Kripke frame of P with K from Definition 2.3.9;

2. f_K is the function defined in 3.2.7.

Definition 3.2.10. Let $\mathcal{O} = (P, K, f, V)$ be a dynamic polyhedral model. Then its *encoding* is $\mathcal{M}(\mathcal{O}) = (\widetilde{K}, \precsim, f_K, \widetilde{V})$, where:

1. $(\widetilde{K}, \precsim, f)$ is the encoding of (P, K, f);

2. $\widetilde{V}(p) = \widetilde{\sigma}$ iff $V(p) \subseteq \widetilde{\sigma}$.

Having this said, we can now formulate and prove a theorem establishing the correspondence between the truth of formulas on $\mathcal{O}$ and on $\mathcal{M}(\mathcal{O})$.

Theorem 3.2.11. Let $\mathcal{O} = (P, K, f, V)$ be a dynamic polyhedral model. Then for every $x \in P$ and every formula φ in $\mathcal{L}_{\gamma,\langle\mathrm{R}\rangle}$ we have

$$\mathcal{O}, x \vDash \varphi \Longleftrightarrow \mathcal{M}(\mathcal{O}), \widetilde{\sigma}^x \vDash \varphi.$$

Proof. The proof is done by induction on the formula φ. The case of propositional letters follows from Definition 3.2.9. Boolean cases follow by the induction hypothesis. Cases of $\Box\varphi$ and $\gamma(\varphi, \psi)$ are completed like in Theorem 2.3.15.

Let us consider the case of $\langle\mathrm{R}\rangle\varphi$.

Assume that $\mathcal{O}, x \vDash \langle\mathrm{R}\rangle\varphi$. Then $x \in f^{-1}([\![\varphi]\!]^{\mathcal{O}})$. By Lemma 3.1.8, and the fact that $[\![\varphi]\!]^{\mathcal{O}} = \bigcup\limits_{\widetilde{\tau} \subseteq [\![\varphi]\!]} \widetilde{\tau}$, we have that $x \in f^{-1}(\widetilde{\tau})$, for some $\widetilde{\tau} \subseteq [\![\varphi]\!]^{\mathcal{O}}$. Then there is $\widetilde{\delta}$ such that $f(\widetilde{\delta}) \subseteq \widetilde{\tau}$, and $x \in \widetilde{\delta}$. Thus, we have that $f_K(\widetilde{\delta}) = \widetilde{\tau}$. Since $\widetilde{\tau} \in [\![\varphi]\!]^{\mathcal{M}(\mathcal{O})}$, by the induction hypothesis, we obtain that $\widetilde{\delta} \in (f_K)^{-1}(\widetilde{\tau}) \subseteq (f_K)^{-1}([\![\varphi]\!]^{\mathcal{M}(\mathcal{O})})$.

For the other direction assume that for $\widetilde{\sigma}^x$ it is the case $\mathcal{M}(\mathcal{O}), \widetilde{\sigma}^x \vDash \varphi$. Then we obtain that $\widetilde{\sigma}^x \in (f_K)^{-1}([\![\varphi]\!]^{\mathcal{M}(\mathcal{O})})$ and $f_K(\widetilde{\sigma}^x) \in [\![\varphi]\!]^{\mathcal{M}(\mathcal{O})}$. Take $\widetilde{\tau}$ such that $f_K(\widetilde{\sigma}^x) = \widetilde{\tau}$. Thus, $f(\widetilde{\sigma}^x) \subseteq \widetilde{\tau}$ and at the same time $\widetilde{\tau} \subseteq [\![\varphi]\!]^{\mathcal{O}}$. Hence, $x \in f^{-1}([\![\varphi]\!]^{\mathcal{O}})$. $\Box$

This theorem, like Theorem 2.3.15, allows us to check the truth of a formula in a dynamic polyhedral model using only the knowledge of the triangulation of this model. Thus, the model checker on dynamic polyhedral model $\mathcal{O} = (P, K, f, V)$ can be carried out using the encoding $\mathcal{M}(\mathcal{O})$.

3.3 Another version of dynamics

In the previous sections, we defined dynamics using a function on a polyhedron P. However, this approach imposes two limitations. First, we cannot model the collapse of a part of the space since the function can only continuously deform the space. For instance, consider the figure 3.1 with four triangles: triangle $T0$ in the middle and three others $T1$, $T2$, $T3$ surrounding the first one on the sides. Assume that we want the middle triangle to disappear, leaving all the other triangles in place. Then, we can define a polyhedral function f on these four triangles so that T1 is sent to some point, say in the middle of his base. But then the function must deform triangles $T1$, $T2$, $T3$, dragging their common border with T0 to the point same point where $T0$ was sent.

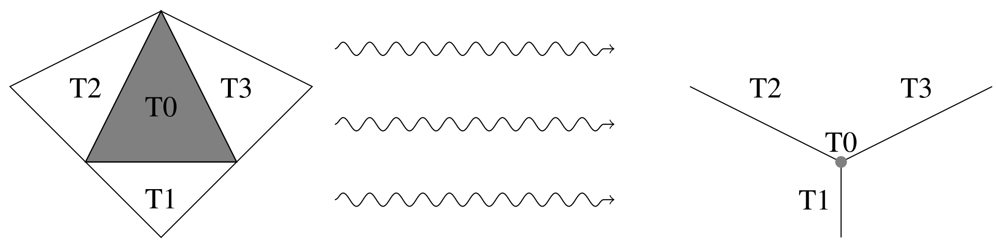

Figure 3.1: Triangle $T0$ is sent entirely to a point - the middle of its base, which entails the contraction of other triangles.

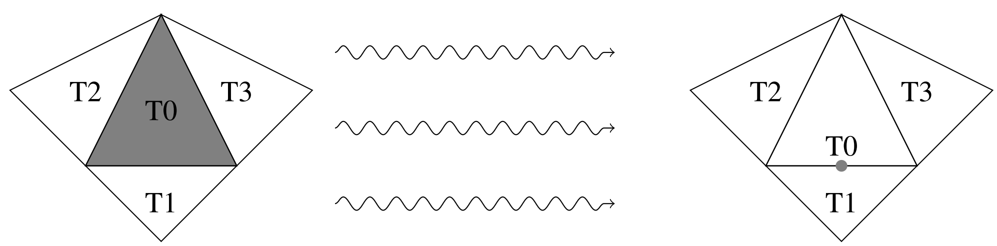

Figure 3.2: Triangle $T0$ goes entirely to the point - the middle of its base, but this time the other triangles may remain in place.

A similar case occurs in the following example. Suppose we have a house with a closed door. Then the simplexes of the door share a common boundary with the simplexes of the wall. Therefore, if we want to model the process of the door being open, we have to define a polyhedral function on the simplexes of both the door and the wall. However, the common boundary must be sent to the common set in this case. Therefore, we will not be able to separate the door's simplexes from the wall's simplexes.

That is why we propose to define a relation instead of a polyhedral function. With this approach, we shall mitigate the conditions and model more flexible dynamics as in the cases 3.2 and 3.3.

That is why the question for the following section is how to define a relation on a polyhedron so that it models dynamics on our polyhedron and simultaneously allows us to implement a model checking algorithm. We suggest the following definition.

Definition 3.3.1. Let P be a polyhedron and K its triangulation. A relation R on P is called *dynamic relation* if:

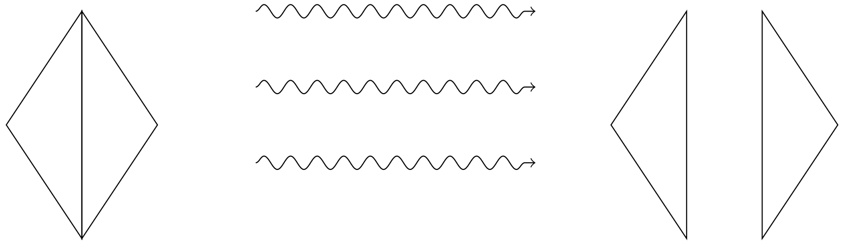

Figure 3.3: Two triangles with a common border can be separated.

1. $R^{-1}(\widetilde{\sigma}) \in \mathfrak{U}(K)$, for every $\widetilde{\sigma} \in \widetilde{K}$;

2. $R^{-1}(P') \in Sub_K(P)$, for every $P' \in Sub_K(P)$.

We tailed the property proved in Lemma 3.1.8 for a polyhedral function into the definition for the dynamic relation.

Lemma 3.3.2. Let P be a polyhedron and K its triangulation. If R is a dynamic relation on P, then xRy and $y \in \widetilde{\sigma}$ implies that $\widetilde{\sigma}^x \subseteq R^{-1}(\widetilde{\sigma})$.

Proof. First, by definition of R we have that $R^{-1}(\widetilde{\sigma}) \in \mathfrak{U}(K)$. Since $\widetilde{\sigma}^x \cap R^{-1}(\widetilde{\sigma}) \neq \varnothing$ we have that there is an element $\widetilde{\delta} \subseteq R^{-1}(\widetilde{\sigma})$ such that $\widetilde{\sigma} \cap \widetilde{\delta} \neq \varnothing$. But then $\widetilde{\sigma}^x = \widetilde{\delta}$ because the relative interiors of different simplexes are disjoint sets. Hence, $\widetilde{\sigma}^x \subseteq R^{-1}(\widetilde{\sigma})$. □

We define the new notion of what *dynamic relational systems* and *dynamic relational models* are.

Definition 3.3.3.

1. We call $\mathcal{D} = (P, K, R)$ a *dynamic relational system* if $\mathcal{D}$ is a dynamic system and R is a dynamic relation.

2. We call $\mathcal{O} = (P, K, R, V)$ a *dynamic relational model* if $\mathcal{O}$ is a dynamic model and R is a dynamic relation.

Observe the following fact.

Lemma 3.3.4. Every dynamic polyhedral system is a dynamic relational system.

Proof. Take $\mathcal{D} = (P, K, f)$ a dynamic polyhedral system. Then by the definition of polyhedral function and Lemma 3.1.8, we have that f is a dynamic relation. Hence, $\mathcal{D}$ is a dynamic relational function. □

For the dynamic model where the relation R is dynamic, we have a lemma analogous to 3.1.10.

Lemma 3.3.5. Let $\mathcal{O} = (P, K, R, V)$ be a dynamic relational model. Then for each element $\widetilde{\sigma} \in \widetilde{K}$, for all points $x, y \in \widetilde{\sigma}$ it is the case that $x \equiv_{\mathcal{L}_{\gamma,\langle\mathrm{R}\rangle}} y$.

Proof. The case of the propositional letter follows from Definition 3.3.2. Cases of boolean operators are trivial. Cases of modalities $\Box$ and $\gamma(\varphi, \psi)$ are the same as in Lemma 2.2.6. And the case for the formula $\langle\mathrm{R}\rangle\varphi$ follows from 3.3.2. $\Box$

Proposition 3.3.6. Let $\mathcal{O} = (P, K, R, V)$ be a dynamic model. Then for any formula φ: $[\![\varphi]\!]^{\mathcal{O}} \in \mathfrak{U}(K)$.

Proof. The proof is analogous to the proof of Proposition 2.2.7. $\Box$

Lemma 3.3.2 also justifies the definition of a relation on $\widetilde{K}$ in the following way.

Definition 3.3.7. Let $\mathcal{D} = (P, K, R)$ be a dynamic relational system. Then define R^* on $\widetilde{K}$ as follows:

$$\widetilde{\sigma} R^* \widetilde{\tau} \text{ iff } \exists x \in \widetilde{\sigma} \exists y \in \widetilde{\tau} : xRy.$$

With R^*, we can now define *dynamic Kripke encoding* for a naïvely monotone dynamic frame.

Definition 3.3.8. Let $\mathcal{D} = (P, K, R)$ be a dynamic relational system. We call $\mathcal{F}(\mathcal{D}) = (\widetilde{K}, \precsim, R^*)$ its *encoding*, if:

1. $(\widetilde{K}, \precsim)$ is the Kripke frame of P and K from Definition 2.3.9;

2. R^* is the relation from 3.3.7.

Definition 3.3.9. Let $\mathcal{O} = (P, K, R, V)$ be a dynamic relational model. We call $\mathcal{M}(\mathcal{O}) = (\widetilde{K}, \precsim, R^*, \widetilde{V})$ its *encoding* if:

1. $(\widetilde{K}, \precsim, R^*)$ is the encoding of (P, K, R);

2. $\widetilde{V}(p) = \widetilde{\sigma}$ iff $V(p) \subseteq \widetilde{\sigma}$.

Definition 3.3.10. Let $\mathcal{F}^D = (W, \sqsubseteq, R)$ be a dynamic Kripke frame. We call R *naïvely monotone* if $x \sqsubseteq yRy'$ implies that $\exists x' : xRx' \sqsubseteq y'$.

Definition 3.3.11. A dynamic Kripke frame $\mathcal{F}^D = (W, \sqsubseteq, R)$ with a naïvely monotone relation we call *naïvely monotone dynamic frame*.

We will show that R^* from Definition 3.3.7 is *naïvely monotone* with respect to relation $\precsim$.

Lemma 3.3.12. Let $\mathcal{D} = (P, K, R)$ be a dynamic relational system and $\mathcal{F}(\mathcal{D}) = (\widetilde{K}, \precsim, R^*)$ its dynamic Kripke counterpart, then R^* is naïvely monotone.

Proof. Assume that $\widetilde{\sigma} \precsim \widetilde{\tau} R^* \widetilde{\tau}'$. Take $R^{-1}(\tau')$. Since τ' is a subpolyhedron in $Sub_K(P)$, $R^{-1}(\tau')$ is also in $Sub_K(P)$. By Lemma 3.1.7 we have that there is downset $C \subseteq \widetilde{K}$ such that $\bigcup C = R^{-1}(\tau)$. Since $\widetilde{\tau} R \widetilde{\tau}'$ we have that $\widetilde{\tau} \cap R^{-1}(\tau') \neq \varnothing$. Then $\widetilde{\tau} \in C$, since C consists of disjoint sets. Since C is downclosed, $\widetilde{\sigma} \in C$. Then $\widetilde{\sigma} \subseteq (R^*)^{-1}(\tau')$. Thus, there is $x \in \widetilde{\sigma}$ and $y \in \tau'$ such that xRy. Take $\widetilde{\sigma}^y$. Since $\widetilde{\sigma}^y \cap \tau' \neq \varnothing$, we have that such that $\widetilde{\sigma}^y \preceq \widetilde{\tau}'$ by Lemma 2.3.12. By definition of R^* we have that $\widetilde{\sigma} R^* \widetilde{\sigma}^y$ and $\widetilde{\sigma}^y \precsim \widetilde{\tau}'$. □

Therefore every encoding for a dynamic relational system is a naïvely monotone dynamic frame.

The above result and Definition 3.3.7 tell us that we can construct a naïvely monotonic relation on $\widetilde{K}$. In the previous section, we constructed a monotone function on $\widetilde{K}$, using a polyhedral function on P. Now we will show how it is possible to construct dynamic relation as in Definition 3.3.1, having naïvely monotonic relation R on $\widetilde{K}$.

Lemma 3.3.13. Let (P, K, R) be a dynamic system with the property $R^{-1}(x) \in \mathfrak{U}(K)$ for every $x \in \mathfrak{U}(K)$. Assume that R' is a relation on $\widetilde{K}$ such that $(\widetilde{K}, \precsim, R')$ is a naïvely monotone Kripke frame with the property:

$$\widetilde{\sigma} R' \widetilde{\tau} \text{ iff } \exists x \in \widetilde{\sigma} \exists y \in \widetilde{\tau} : xRy$$

Then $R^{-1}(P') \in Sub_K(P)$ for every $P' \in Sub_K(P)$.

Proof. Take $P' \in Sub(P)$. Then $P' = \bigcup C$ for some downset $C \subseteq \widetilde{K}$ by Lemma 3.1.7. We will show that $R^{-1}(\bigcup C) = \bigcup D$, for some closed $D \subseteq \widetilde{K}$. This will conclude the proof by lemma 3.1.7. First observe that $R^{-1}(\bigcup C) \in \mathfrak{U}(K)$. Take $D = \{\widetilde{\sigma} \mid \widetilde{\sigma} \subseteq R^{-1}(\bigcup C)\}$. Then $\bigcup D = R^{-1}(\bigcup C)$. Let us show that D is closed. Suppose $\widetilde{\sigma}_1 \in D$ and $\widetilde{\sigma}_2 \precsim \widetilde{\sigma}_1$. Since $\widetilde{\sigma}_1 \subseteq R^{-1}(\bigcup C)$, we have that there is $\widetilde{\tau}_1 \in C$ with $y \in \widetilde{\tau}_1$ such that for some $x \in \widetilde{\sigma}_1$: xRy. Thus, $\widetilde{\sigma}_1 R' \widetilde{\tau}_1$. By naïve monotonicity of R' we have that there is $\widetilde{\tau}_2$ such that $\widetilde{\sigma}_2 R' \widetilde{\tau}_2 \precsim \widetilde{\tau}_1$. Since C is closed, $\widetilde{\tau}_2 \in C$. Then we have that $\widetilde{\sigma}_2 \cap R^{-1}(\bigcup C) \neq \varnothing$, and from this follows that there is some $\widetilde{\delta} \subseteq R^{-1}(\bigcup C)$ such that $\widetilde{\sigma}_2 \cap \widetilde{\delta} \neq \varnothing$. Hence, $\widetilde{\sigma}_1 = \widetilde{\delta}$, due to their disjointness, and $\widetilde{\sigma}_2 \subseteq R^{-1}(\bigcup C)$. □

When R is a function, we have that $R^{'}$ is also a function, and therefore naïve monotonicity becomes just monotonicity.

We finalize the section with the correspondence for truth between formulas in models.

Theorem 3.3.14. Let $\mathcal{O} = (P, K, V)$ be a dynamic relational model. Then for every $x \in P$ and every formula φ in $\mathcal{L}_{\gamma,\langle\mathrm{R}\rangle}$ we have

$$\mathcal{O}, x \vDash \varphi \Longleftrightarrow \mathcal{M}(\mathcal{O}), \widetilde{\sigma}^x \vDash \varphi.$$

Proof. The proof is done by induction on the formula φ. The case of propositional letters follows from Definition 3.2.9. Boolean cases follow by the induction hypothesis. Cases of $\Box\varphi$ and $\gamma(\varphi, \psi)$ are completed like in Theorem 3.2.11.

Let us consider the case $\langle\mathrm{R}\rangle\varphi$.

($\Rightarrow$) Assume that $\mathcal{O}, x \vDash \langle\mathrm{R}\rangle\varphi$. Then $x \in R^{-1}([\![\varphi]\!]^{\mathcal{O}})$. Since $R^{-1}([\![\varphi]\!]^{\mathcal{O}}) \in \mathfrak{U}(K)$, then there is an $\widetilde{\sigma}$ such that $x \in \widetilde{\sigma} \subseteq R^{-1}([\![\varphi]\!]^{\mathcal{O}})$. Take $y \in [\![\varphi]\!]^{\mathcal{O}}$ and $\widetilde{\sigma}^y$ such that xRy. By definition of $R^* : \widetilde{\sigma} R^* \widetilde{\sigma}^y$. By inductive hypothesis we have that $\mathcal{M}(\mathcal{O}), \widetilde{\sigma}^y \vDash \varphi$. Hence, $\mathcal{M}(\mathcal{O}), \widetilde{\sigma} \vDash \langle\mathrm{R}\rangle\varphi$.

($\Leftarrow$) If $\mathcal{M}(\mathcal{O})\widetilde{\sigma}^x \vDash \langle\mathrm{R}\rangle\varphi$, then there is $\widetilde{\tau} \in [\![\varphi]\!]^{\mathcal{M}(\mathcal{O})}$ such that $\widetilde{\sigma}^x R^* \widetilde{\tau}$. Therefore, there is $z_1 \in \widetilde{\sigma}^x$ and $z_2 \in \widetilde{\tau}$ such that $z_1 R z_2$. By lemma 3.3.2, we have that there is some $y \in \widetilde{\tau}$ such that xRy. Hence, $\mathcal{O}, y \vDash \varphi$ by the inductive hypothesis. We conclude that $\mathcal{O}, x \vDash \langle\mathrm{R}\rangle\varphi$. $\Box$

We conclude this chapter by summing up what has been done.

- We define a new language $\mathcal{L}_{\gamma,\langle\mathrm{R}\rangle}$ for reasoning about dynamic structures.
- We defined the polyhedral function and dynamic relation, which allowed us to define novel notions of *dynamic polyhedral models* and *dynamic relational models*;
- For both dynamic polyhedral models and dynamic relational models, we defined how to construct their encodings;
- We proved that a point x in dynamic polyhedral model $\mathcal{O}$ satisfies formula φ if an only if $\widetilde{\sigma}^x$ satisfies φ in its encoding $\mathcal{M}(\mathcal{O})$. The same result was obtained for dynamic relational models and their encodings.

Chapter 4

Polyhedral model checking

In this chapter, we will focus on model checking. We will formulate the problem that model checking solves and explain how the model checking algorithm can be applied to 3D models. Then we will provide the algorithm for model checking on the encoding of dynamic polyhedral models. We then demonstrate the application of the PolyLogicA model checking with a new example from the architectural domain. We conclude the section with a conceptual description of a prototype dynamic model checking software.

4.1 Model checking algorithm

Model checking is a formal verification method for software systems that is used to identify errors in programs and protocols. It involves: 1. creating a mathematical model $\mathcal{M}$ representing system states and 2. specifying the property with a formula φ formulated in a formal language. The key advantage of model checking is that it is fully automated.

In Chapter 2, we showed that it is possible to consider 3D models instead of classical software examples by representing them as polyhedra equipped with a triangulation. We defined our mathematical model as $\mathcal{E} = (P, K, V)$ in Definition 2.2.3. By employing model checking, we want to verify some properties. For instance, "reachability of the exit in the building" in the context of a 3D model of a building. It was shown in Theorem 2.3.15 that this question can be reduced to the question of checking the property on an encoding $\mathcal{M}(\mathcal{E}) = (\widetilde{\mathcal{K}}, \precsim, \widetilde{V})$ of $\mathcal{E}$. This method was developed in the project `PolyLogicA`, where it is used for region segmentation of a 3D model based on triangulation.

In our case, we consider dynamic 3D models. Adding dynamics to a polyhedron allows us to check the properties of the form "it will be possible to reach the exit from the building at the next moment." There is a critical issue of a tradeoff between the possibilities of dynamics and the efficiency of our model checking algorithm. In general, if we consider some dynamic functions on our polyhedron (e.g., an affine transformation), there may be no correspondence between a dynamic model and a finite dynamic Kripke frame, which is the method of model checking that we employ. But since we use the definition of dynamic relations such as dynamic polyhedral function (Definition 3.1.6) and dynamic relation (Definition 3.3.1), for us checking the truth of a formula at a point x in a dynamic polyhedral model $\mathcal{O}$ (or a dynamic relational model) reduces to the problem of checking the truth of a formula on the point $\widetilde{\sigma}^x$ of its encoding $\mathcal{M}(\mathcal{O})$, which is a finite dynamic Kripke frame (Theorems 3.2.11 and 3.3.14).

Since we have two different definitions types of dynamic models, which are dynamic polyhedral models (Definition 3.1.9) and dynamic relational models (Definition 3.3.3), we have to define algorithms for both of their encodings, which are finite dynamic monotone Kripke models and finite dynamic naïvely monotone Kripke models respectively. However, since every dynamic monotone Kripke model is also a dynamic naively monotone model, we will formulate the model checking algorithm for dynamic naively monotone Kripke models.

We provide an algorithm for computing the truth set of formulas

$$p \mid p \wedge q \mid \neg p \mid \Diamond p \mid \gamma(p, q) \mid \langle \mathrm{R} \rangle p \tag{4.1}$$

This will give us an algorithm for finding a truth set of all formulas because, for example, we can calculate $[\![\langle \mathrm{R} \rangle \varphi]\!]$ by treating $[\![\varphi]\!]$ as an interpretation of the propositional variable p_φ and then apply the algorithm to the formula $\langle \mathrm{R} \rangle p_\varphi$.

So, our model checking algorithm will take as input the encoding $\mathcal{M}(\mathcal{O}) = (\widetilde{K}, \precsim, R^*, \widetilde{V})$, and formula φ from (4.1). The output will be the set $Sat(\varphi) = \{x \mid \mathcal{M}(\mathcal{O}), x \vDash \varphi\} = [\![\varphi]\!]^{\mathcal{M}(\mathcal{O})}$ of nodes in $\mathcal{M}(\mathcal{O})$ that satisfy formula φ. Thus, our algorithm is a so-called "global" model checking algorithm, as opposed to other methods that only check satisfaction at a single point. The algorithm for Boolean combinations is straightforward and is thus omitted. The algorithm for $\Diamond$ is identical to the algorithm for $\langle \mathrm{R} \rangle$, but instead of the R^* relation, we use the relation $\sqsubseteq$. Thus, we present two algorithms: one for modality γ and the other one for modality $\langle \mathrm{R} \rangle$. The algorithm for γ is borrowed from [Bez+22].

We use pseudocode to define the algorithms. Using **Require**, we denote the input data for the algorithm. With **Ensure** we denote the output. With notation `var:= <expression>` we will denote the assignment of the value `<expression>` to the variable `var`. Since there are two relations $\sqsubseteq$ and R in our model, we use $\mathtt{out}^{\cdot}(x) = \{y \mid y \cdot x\}$ to denote the set of all predecessors of x, where $\cdot \in \{\sqsubseteq, R\}$. Similarly for $\mathtt{in}^{\circ}(x) = \{y \mid x \circ y\}$, where $\circ \in \{\sqsubseteq, R\}$.

Algorithm 1 Algorithm for computing $\langle \mathrm{R} \rangle \varphi$

Require: $\mathcal{M}(\mathcal{O}) = (\widetilde{K}, \precsim, R^*, V)$, $\langle \mathrm{R} \rangle p$
Ensure: $[\![\langle \mathrm{R} \rangle p]\!]^{\mathcal{M}}$

$SemP := [\![p]\!]^{\mathcal{M}}$
$res := \varnothing$
for $x \in SemP$ **do**
    $predStates := \mathtt{out}^{R}(x)$
    $res := predStates \cup res$
end for
return: res

Algorithm 2 Algorithm for computing $\gamma(\varphi, \psi)$

Require: $\mathcal{M}(\mathcal{O}) = (\widetilde{K}, \precsim, R^*, V)$, $\gamma(p, q)$
Ensure: $[\![\gamma(p, q)]\!]^{\mathcal{M}}$

$frontier := [\![p]\!]^{\mathcal{M}} \cap \mathtt{out}^{\sqsubseteq}([\![q]\!]^{\mathcal{M}})$
$flooded := frontier$
while $frontier \neq \varnothing$ **do**
    $x = frontier.pop()$
    for $y \in \mathtt{in}^{\sqsubseteq}(x) \cup \mathtt{out}^{\sqsubseteq}(x)$ **do**
        if $y \notin flooded \;\&\; y \in [\![p]\!]^{\mathcal{M}}$ **then**
            $frontier.add(y)$
            $flooded.add(y)$
        end if
    end for
end while
$res := \mathtt{in}(flooded)$
return: res

Correctness of the algorithm 1 is trivial and follows directly from its definition and semantics of $\langle \mathrm{R} \rangle p$. As for the algorithm 2, the proof for its correctness is sketched below and can be found in [Bez+22].

Correctness, sketch A path $\pi : \{0, \dots, k\} \to \widetilde{K}$ we call a "good" path if it starts at a point that witnesses the satisfaction of $\gamma(p, q)$. Observe that a good path can be divided into three parts, namely the beginning of the path $\pi(0)$, the middle of the path $\pi(\{1, \dots, k-1\}) \subseteq [\![p]\!]^{\mathcal{M}}$, and the end of the path $\pi(k) \in [\![q]\!]^{\mathcal{M}}$. We will work backwards, starting from the set of points that satisfy p and are connected to a node satisfying q. So, first we compute the set $C := [\![p]\!]^{\mathcal{M}} \cap$ out$([\![q]\!]^{\mathcal{M}})$. C corresponds to all the nodes satisfying $\pi(k-1)$ in some good path π. Then we use the flooded procedure to build the set D of the nodes of the graph that are connected to C with a non-directed path, i.e. while choosing the nodes, we abstract from the direction of the edges. This part corresponds to all nodes $\pi(j)$ for $j = 1, \dots, k-1$, for any good path π. Finally, we compute the set $[\![\gamma(p, q)]\!]^{\mathcal{M}} = \mathtt{in}(D)$, which are all the nodes that correspond to the initial nodes of a good path.

Complexity We indicate by n the number of nodes in $\widetilde{K}$ and by d the dimension of the corresponding polyhedron. First, consider the complexity bound for Algorithm 1. Since there are n elements in the domain of $\mathcal{M}(\mathcal{O})$, the number for R -predecessors for each element is at most n. Hence, the asymptotic computational complexity of Algorithm 1 is of order $\mathcal{O}(n^2)$.

Remark 4.1.1. We note that in the case of so-called "snapshot models" [Kon+07], the complexity could be reduced, which will be explored in future work (see section 4.2).

Considering the complexity of algorithm 2, observe that every node x has at most $2^{d_x+1} - 1$ edges (where d_x is the dimension of simplex x) since each simplex can have at most $2^{d_x+1} - 1$ proper faces. Let us denote with N the sum of nodes and edges in $\mathcal{M}(\mathcal{O})$. Then, N for $\mathcal{M}(\mathcal{O})$ is $n \cdot 2^{d+1}$. This number grows exponentially in d if d is not fixed. However, since we work with 3D models, we fix $d = 3$. Therefore, the contribution of d to the encoding of the model becomes a constant, and the size of the encoding of $\mathcal{M}(O)$ is of order $\mathcal{O}(n)$ in this case. The flooding procedure in Algorithm 1 has linear computational complexity in the number of nodes and edges of $\mathcal{M}(\mathcal{O})$, that is, N. Again, since every subformula of $\gamma(\varphi, \psi)$ is checked independently from each other, the asymptotic computational complexity of the model checking algorithm is of order $\mathcal{O}(N \cdot h)$, and once the dimension is fixed it becomes $\mathcal{O}(n)$.

The computation of Boolean operators is also linear in N. That is why the model checking algorithm for them is also of order $\mathcal{O}(n)$, independently of the dimension.

We conclude that the total complexity of the spacial model checking algorithm for a fixed dimension d ($d = 3$ in our case) is of polynomial complexity.

4.2 The application of model checker

In the setting of our work, we aimed to present a theoretical approach that would allow us to consider, in principle, the problem of analyzing dynamic 3D models. Developing a prototype for such a model checker would require separate research. This section aims to apply the model checker `PolyLogicA` to a real-world scenario. Before `PolyLogicA` was tested on two examples: a 3D cube and an existing 3D model from the medical domain. In this work, we extend its application possibilities and try it on a new example from the architectural domain. It required the modification of the source code of the tool to prepare the data. In the second part of this section, we delve into the elucidation of the prototype for dynamic model checking.

4.2.1 Model checker for static 3D models

The authors of [Bez+22] have developed a prototype of a model checker for static 3D models. The tool `PolyLogicA` was presented in [Bez+22] and is currently available at `github` repository[1]. `PolyLogicA` is Free and OpenSource Software distributed under the Apache 2.0 licence.

Specification of `PolyLogicA` 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.

Let us closely consider the `load` command. It allows a user to load a 3D model. The 3D model is specified using `.json`[2] file. The information contained in the file consists of:

1. A list p of d-dimensional vectors, denoting the coordinates of the 0-cells of the polyhedron;

2. A list of atomic propositions;

[1] See https://github.com/vincenzoml/VoxLogicA.

[2] See https://www.json.org/.

3. A list of simplexes.

Each simplex σ is described using the indexes of its vertices in p. The specification of σ contains the list of atomic propositions $p_1, \ldots, p_n$ that hold in $\widetilde{\sigma}$.

After specifying which file has to be loaded, the user has to declare the variables and their combinations using `let`. The file is concluded with the `save` statement.

Data preparation Let us now take a closer look at a 3D model we will use. In our running example, we consider the *existing* model of a villa to demonstrate vast application possibilities of `PolyLogicA`. To open the file of a 3D model, we will use the software MeshLab developed by the Visual Computing Lab of ISTI-CNR [Cig+08]. The initial model can be found in Figure 4.1. This is essentially our polyhedron. First, we fix its triangulation. This can be done using the function from MeshLab:

Filters $\rightarrow$ Remeshing, Simplification and Reconstruction $\rightarrow$
$\rightarrow$ Simplification: Quadric Edge Collapse Decimation

This function uses the simplification algorithm of Garland and Heckbert presented in [GH97]. The result is depicted in Figure 4.2. Once we used the triangulation, we obtain a so-called 3D mesh of a model. The universal problem that one might have while working with a 3D model and using `PolyLogicA` is that the neighbourhood simplexes of the 3D mesh might not be connected to each other. Whether simplices are connected can be checked using the tool **Select a connected component in the region**. Disconnectedness can cause problems during the experiment phase; since we use the reachability operator γ, we need to have (almost) the entire 3D mesh to be connected. Thus, to eliminate this possibility, we can use another function from MeshLab:

Filters $\rightarrow$ Cleaning and Repairing $\rightarrow$ Merge Close Vertices

Conversion to PolyLogicA Once this process is done, we have to convert the file with the 3D mesh to `.json` format to load it to the model checker. This conversion involved writing a new version of the `python`[3] program, which imports the `.obj` meshes in PolyLogicA. Every `.obj` file has several sections. We were interested in two of them: the one that specifies vertices and the one that specifies the faces of

[3] See https://www.python.org/

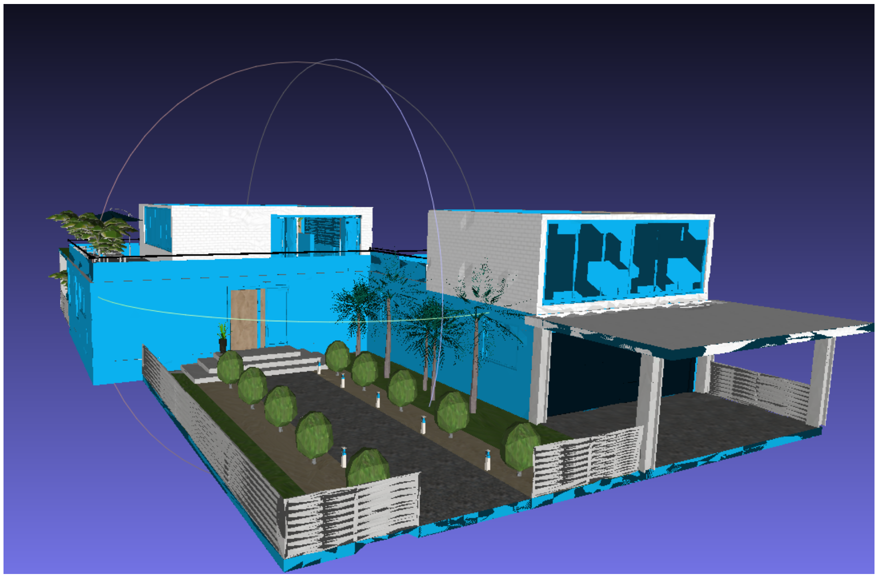

Figure 4.1: The initial 3D model of a building.

3-dimensional simplexes. Along with this data, we also had to specify the propositions we would assign to the relative interiors of simplexes. In our real-world 3D meshes, each simplex is associated with some material (e.g. wood, stone, etc.). The program can be found in the fork of the VoxLogicA at our GitHub repository[4] under the name `program.py`.

Model checking Once the file has been parsed, we can write the text file that will specify the task for our model checker. Figure 4.3 illustrates an example of such a file. The code is self-explanatory, and we will explain only operations `near` and `through`. Operator `near` stands for taking the topological closure of simplexes. We use it to have all the simplexes (i.e. triangles, intervals, and vertices) that satisfy given variables. Operator `trough` stands for γ operator. So, our final variable `house` denotes all the points on the wall or floor such that it is possible to reach the floor from them by passing only through the wall.

In addition to the polyhedra model checker, a visualizer was also presented in [Bez+22]. It takes as input the `.json` file with the loaded model and with the output of `PolyLogicA` and outputs the visualization of the result. In our case, the visualization of the query formulated in 4.3 can be found in Figure 4.4. One can see that the result precisely depicts the areas that we wanted to separate.

[4] https://github.com/Fgrtue/VoxLogicA/tree/polyhedra/src

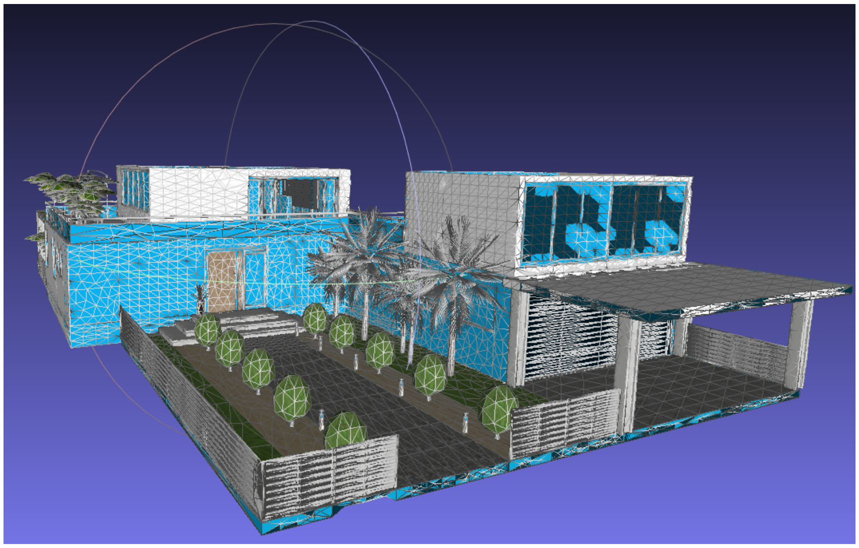

Figure 4.2: Trinagulated model

In summary, the skeleton of the procedure is as follows:

1. Take a 3D model and triangulate it;
2. Check whether the regions that you want to run your query on are connected;
3. Parse the model in order to obtain a `.json` file with simplices and materials for it;
4. Write the `.json` file identifying the region that is intended to be extracted;
5. Inspect the visualization of the result using the visualizer.

4.2.2 Outlook: efficient model checking for dynamic 3D models.

In this subsection, we present how the theory we developed in the previous chapters can be applied to building a prototype for a model checker of dynamic models. The definition of dynamic systems we provided in Chapter 3 underpins a more theoretical view of the dynamics rather than a view of applications. This is because the model $\mathcal{O} = (P, K, R, V)$ is a monolith in which the entire dynamical component of the model is hidden in the relation R. Whereas in reality, when modelling processes, we deal with a discrete set of states, each of which reflects some particular state of the model. From this point of view, real dynamics resembles Snapshot

```
load model = "./Kirill.ThesisExample.02/house 2/
   source/House/villa_small.json"

//Atomic propositions for floor and wall in the house
let floor = ap("material_58")
let wall = ap("material_1")

//Taking closure of the defined propositions
let allfloor = near(floor)
let allwall = near(wall)

//Q: separate the house
let house = through(allwall,allfloor) & (or(allwall,
   allfloor))

save "house'' house
```

Figure 4.3: An example of an input file for PolyLogicA. The first line loads the model from a directory on our machine. Then we specify the propositions, which are `floor` and `wall` in our model. They correspond to `material`$_{5}$8 and `material`$_{1}$ respectively. An important detail was that the materials were assigned only to the relative interiors of 2-simplexes, not 1 and 0-simplexes. Therefore, to consider all the simplexes, we have to use operator `near`, which semantically works as closure. After that, we can apply the reachability operator `trough` to `allwall` and `allfloor`, which gives us all the points that can reach the floor from the wall.

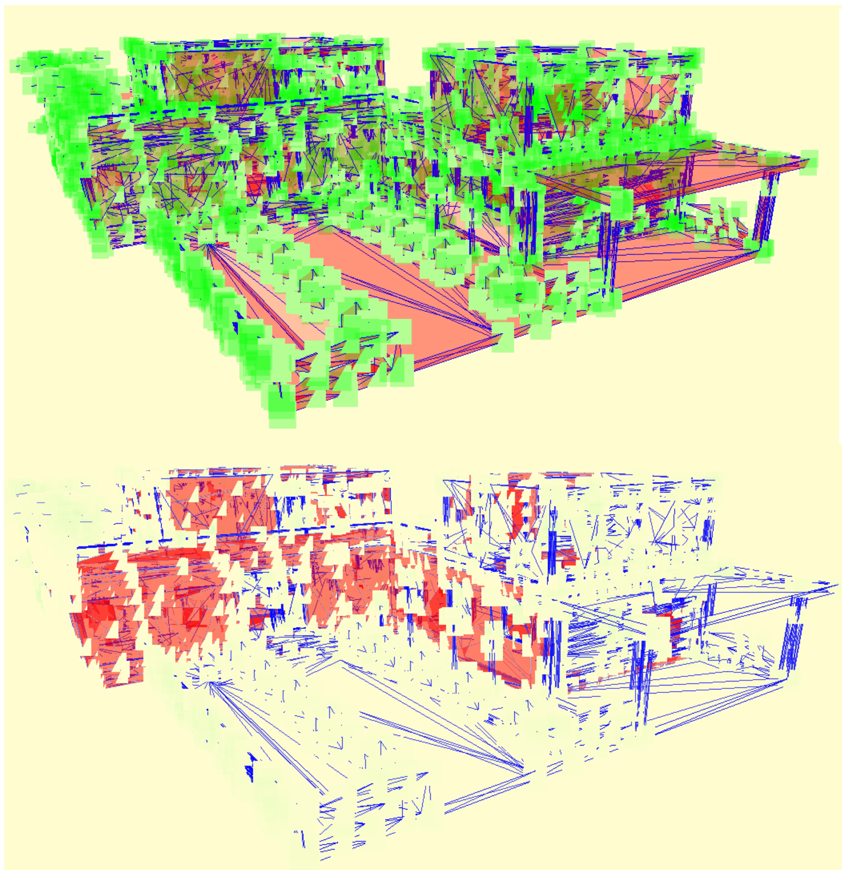

Figure 4.4: The top image visualizes the model of the villa in the visualizer of PolyLogicA, without application of the query from Figure 4.3. The green squares denote 0 - simplexes, blue lines denote 1 - simplexes, and red planes denote 2 - simplexes. As we can see, the villa model contains two blocks of building, an adjoining territory in front of the left block, and a little patio in front of the right block. The query in Figure 4.3 aims to separate the two blocks, excluding the adjoining territory and the patio. The image on the bottom shows a picture after applying the query from Figure 4.3. The red planes denote the result of the query. We hide the 0-simplexes so that they do not distract us from the result. However, visualizer leaves empty space in their places. Overall, we can see that the model checker has extracted exactly two blocks of the building as wanted.

models [Kon+07]: $\mathcal{M}_0, \mathcal{M}_1, \ldots$, i.e. a sequence of static models, each of which represents some state of the model. Thus, the verification of dynamic properties of a dynamic 3D model will be the verification of a succession of properties in different moment-states of that model. In this sense, structurally, the future algorithm for the prototype should be similar to the algorithm used in PolyLogicA. The initial research on building a prototype for a dynamic model checker could be to translate our dynamic models into Snapshot models and show that the model checker defined in this work will be sufficient to check the properties of snapshot models.

The next important aspect of the technical realization of a prototype is related to the file format in which the 3D model is written. The dynamic of a 3D model can be simulated in various applications with built-in physics and simulation features. For instance, Unity[5] and Blender[6] are such applications. Files that specify dynamic models are stored in formats `.blend`, `.fbx` and `.dae`, each of which represents a 3D model differently. Thus, this line of research would be related to extracting the model, and different model states, from these formats to represent them in a single file (possibly in `.json` format) that would be loaded into the model checker.

Defining the specification for the file `input` to the model checker is necessary. In general, this file should not be very different from the one for `PolyLogicA`; the only difference should be the introduction of a new syntax for temporal modalities. In our work, we extended the language $\mathcal{L}_{\Box,\gamma}$ only with one new temporal modality $\langle \mathrm{R} \rangle$ (next). However, it would make sense to consider modalities like $[\exists]$ (eventually) and $[\forall]$ (always in the future) since they give much more possibilities for expressing the future states of the model.

Finally, the last but essential step would be to create a visualizer for the results. It should take as input a `.json` file containing the model and the result of our query. Since we have several snapshots of a 3D model, the visualizer should show different static states of the model at which we will see the result of the query. As the highlighted region will differ for each step, we can track how it has changed. In this way, the visualization would show which parts of the model are highlighted depending on the temporal state of the model.

We conclude this chapter by summing up what has been done:

- Defined the model checking algorithm, proved its correctness, and found the upper bound for its complexity;

- Explained in detail how the prototype `PolyLogicA` works on a new-case scenario from the architectural domain;

[5] See https://unity.com/.
[6] See https://www.blender.org/.

- Outlined the construction methodology of a prototype for a dynamic model checker.

Chapter 5

Completeness of dynamic logics

In this chapter, we present an investigation of two dynamic logics within the language $\mathcal{L}_{\Box,\langle\mathrm{R}\rangle}$ and establish their completeness. The chapter is organized as follows.

In the first section, we introduce the concept of logics, the validity of formulas, and completeness, formulate two logics $\mathcal{DPL}$ and $\mathcal{DRL}$ and, in the last three sections, we focus on proving their completeness. Our strategy for these proofs involves the combination of two techniques outlined in [FM21], [Ada19], and [Ada+22]. To ensure the proof's clarity, we thoroughly explain the intuition behind these techniques.

The first technique centers around the finite model property of dynamic logics. We achieve this by manipulating a canonical model of the logic. The process begins by defining a finitary relation $\sqsubseteq_\Phi$, which depends on a finite set Φ and is established in a manner similar to selective filtration (see [Cha97]). Utilizing this relation, we recursively extract a finite structure known as a "moment" for every point w of the canonical model. These moments capture the 'static' information of w. We then build a dynamic model by defining a procedure for constructing the next moment from a given one. This allows us to generate a sequence of moments starting from a single point, with each moment connected to its predecessor through a dynamic relation. We refer to such a sequence as a "story". Importantly, every story is finite and contains static and dynamic information about a point.

The second technique involves constructing a dynamic system from a dynamic Kripke frame, specifically a polyhedron equipped with a relation. This technique uses the "nerve" of a poset, which consists of the collection of finite chains in the poset ordered by inclusion. The construction of the dynamics on the polyhedron is the most complicated aspect and involves several steps to achieve it. This polyhedron will satisfy the formula that is satisfied on the canonical model. The construction details are in section 5.4.

5.1 Dynamic logics $\mathcal{DPL}, \mathcal{DRL}$

We will concentrate solely on the dynamic part of our systems, i.e. we restrict our language to $\mathcal{L}_{\Box,\langle \mathrm{R}\rangle}$. Observe that this restriction does not affect the definitions and results established in the previous chapter since modalities $\Box$ and $\langle \mathrm{R}\rangle$ are independent of γ. First, we define the validity of formulas in our dynamic systems and frames. Then we abstract away from the structures we are working with and define what logic is. We give two examples of logics and prove their consistency concerning the corresponding class of structures.

We will now define the validity of formulas.

Definition 5.1.1.

1. For a dynamic system $\mathcal{D} = (\mathrm{P}, \mathrm{K}, R)$ and formula φ, we say that φ is *valid on* $\mathcal{D}$ (notation: $\mathcal{D} \models \varphi$), whenever $(\mathcal{D}, V) \models \varphi$ for all valuations V on $\mathcal{D}$.

2. For a class of dynamic systems $\mathbf{D}$ and a formula φ we will say that φ is *valid on* $\mathbf{D}$ (notation: $\mathcal{D} \models \varphi$), whenever for each dynamic system $\mathcal{D}$ in $\mathbf{D}$: φ is valid on $\mathcal{D}$.

3. For a class $\mathbf{D}$ of dynamic systems, a set of formulas $\mathcal{L}(\mathbf{D}) = \{\varphi \mid \mathbf{D} \models \varphi\}$ is called the logic of $\mathbf{D}$.

So far, we have been working within two types of dynamic systems: dynamic polyhedral functions and dynamic relations. Let us formulate this in a mathematically rigorous way.

Definition 5.1.2. Let $\mathbf{D}$ be a class of dynamic systems. Then:

1. We call $\mathbf{D}$ a class of dynamic polyhedral systems (notation: $\mathbf{DPS}$) if for every $\mathcal{D} = (\mathrm{P}, \mathrm{K}, R)$ in $\mathbf{D}$ we have that R is dynamic polyhedral function.

2. We call $\mathbf{D}$ a class of dynamic polyhedral systems (notation: $\mathbf{DRS}$) if for every $\mathcal{D} = (\mathrm{P}, \mathrm{K}, R)$ in $\mathbf{D}$ we have that R is dynamic relation.

We also introduce validity for dynamic Kripke frames:

Definition 5.1.3.

1. For a dynamic Kripke frame $\mathcal{F}^d = (\mathrm{W}, \sqsubseteq, R)$ and a formula φ, we will say that φ is *valid on* $\mathcal{F}^d$ (notation: $\mathcal{F} \models \varphi$), whenever $(\mathcal{F}^d, V) \models \varphi$ for every valuation V on $(\mathrm{P}, \mathrm{K}, R)$.

2. For a class of dynamic Kripke frames $\mathbf{F}$ and a formula φ we will say that φ is *valid on* $\mathbf{F}$ (notation: $\mathbf{F} \vDash \varphi$), whenever for each $\mathcal{F}$ from $\mathbf{F}$ φ is valid on $\mathcal{F}$.

3. For a class $\mathbf{F}$ of dynamic Kripke frames, a set of formulas $\{\varphi \mid \mathbf{D} \vDash \varphi\}$ is a logic of $\mathbf{F}$.

Now we define the notion of logic.

Definition 5.1.4. A *substitution* σ is a map $\sigma \colon Var \to Fm$. A substitution σ is extended to a *uniform substituiton* $\overline{\sigma} \colon Fm \to Fm$ by recursion:
1. $\overline{\sigma}(p) = \sigma(p)$, for all $p \in Var$;
2. $\overline{\sigma}(\neg\psi) = \neg\overline{\sigma}(\psi)$;
3. $\overline{\sigma}(\psi \wedge \gamma) = (\overline{\sigma}(\psi) \wedge \overline{\sigma}(\gamma))$;
4. $\overline{\sigma}(\Box\psi) = \Box\overline{\sigma}(\psi)$;
5. $\overline{\sigma}(\langle \mathrm{R} \rangle\psi) = \langle \mathrm{R} \rangle\overline{\sigma}(\psi)$;

Definition 5.1.5. We call *logic* L a set of formulas Fm in language $\mathcal{L}_{\Box,\langle \mathrm{R} \rangle}$, that contains axioms:

1. all the tautologies;

2. formula $\Box(p \wedge q) \leftrightarrow (\Box p \wedge \Box q)$ (we denote it with $\mathrm{K}_\Box$);

3. formula $(\langle \mathrm{R} \rangle p \vee \langle \mathrm{R} \rangle q) \leftrightarrow \langle \mathrm{R} \rangle(p \vee q)$ (we denote it with $\mathrm{K}_{\langle \mathrm{R} \rangle}$);

4. formula $\Box(\Box(p \to \Box p) \to p) \to p$ (we denote it with $\mathbf{grz}$),

and is closed under the rules:

1. (MP) $\left(\dfrac{\varphi \to \psi, \varphi}{\psi}\right)$;

2. (Uniform) $\left(\dfrac{\varphi}{\overline{\sigma}(\varphi)}\right)$, where $\overline{\sigma}$ is uniform substitution

3. $(\mathrm{Nec}_\Box)\left(\dfrac{\varphi}{\Box\varphi}\right)$

4. $(\mathrm{Nec}_{[\mathrm{R}]})\left(\dfrac{\varphi}{[\mathrm{R}]\varphi}\right)$

For a formula $\varphi \in \mathsf{L}$ we say that φ is a theorem of L, and denote it with $\vdash_{\mathsf{L}} \varphi$.

Definition 5.1.6. For a set Γ, formula φ and logic L we say that φ is *deducible in* L *from* Γ if $\vdash_{\mathsf{L}} \varphi$ or there are formulas $\psi_1, \ldots, \psi_n \in \Gamma$ such that:

$$\vdash_{\mathsf{L}} (\psi_1 \wedge \ldots \wedge \psi_n) \rightarrow \varphi.$$

If this is the case we write $\Gamma \vdash_{\mathsf{L}} \varphi$, if not, $\Gamma \nvdash_{\mathsf{L}} \varphi$. A set of formulas Γ is L-consistent if $\Gamma \nvdash_{\mathsf{L}} (\varphi \wedge \neg\varphi)$ for every formula ϖ. Otherwise we call Γ an L -inconsistent.

Remark 5.1.7. Recall that $\mathbf{K}$ is the smallest monomodal normal logic, and $\mathbf{Grz} = \mathbf{K} + \mathbf{grz}$.

We can consider different logics. All of them will describe the behaviour of two modalities: $\Box$ and $\langle \mathrm{R} \rangle$. The smallest logic we denote with $\mathbf{Grz} \otimes \mathbf{K}$, which is the basic fusion of monomodal logic $\mathbf{Grz}$ for $\Box$ and monomodal logic $\mathbf{K}$ for $\langle \mathrm{R} \rangle$. To denote the logics that extend $\mathbf{Grz} \otimes \mathbf{K}$ we use the following naming conventions: logic

$$\mathbf{X}\text{-}\mathbf{Y}\text{-}\mathbf{Z}_1\text{-}\ldots\text{-}\mathbf{Z}_n$$

is the smallest logic that extends the fusion $\mathbf{X} \otimes \mathbf{Y}$ of logic $\mathbf{X}$ for modality $\Box$ and $\mathbf{Y}$ for modality $\langle \mathrm{R} \rangle$ with bimodal interaction axioms $\mathbf{Z}_1, \ldots, \mathbf{Z}_n$.

Now we discuss the properties of logics that extend **Grz** - **K**.

Definition 5.1.8. Relation $\leq$ is called Noetherian if there are no infinite paths $x_0 \leq x_1 \leq x_2 \leq \ldots$ such that for all i we have $x_i \neq x_{i+1}$.

Lemma 5.1.9. Let $\mathcal{F} = (\mathrm{W}, \sqsubseteq)$ be a Kripke frame. Then $\mathcal{F} \vDash \boldsymbol{grz}$ iff $\sqsubseteq$ is reflexive, transitive and Noetherian.

Proof. See [Cha97, Proposition 3.48]. $\Box$

Definition 5.1.10. Let $\mathcal{F}^d = (W, \sqsubseteq, R, V)$ be a dynamic Kripke frame. We will say that $\mathcal{F}^d$ is a $\mathbf{Grz}$ Kripke frame, whenever $\mathcal{F}^d \vDash \mathbf{Grz}\text{-}\mathbf{K}$.

Recall that for every dynamic polyhedral model (and dynamic relational model) $\mathcal{O} = (P, K, R, V)$ there is an encoding $\mathcal{M}(\mathcal{O}) = (\widetilde{K}, \precsim, R^*, \widetilde{V})$. In the previous section, in Theorems 3.2.11 and 3.3.14, we proved that a formula is satisfied at a point of a dynamic polyhedral and relational model $\mathcal{O}$ iff it is satisfied in its encoding $\mathcal{M}(\mathcal{O})$. Now we will leverage this result to the validity of formulas on dynamic polyhedral and dynamic relational systems $\mathcal{D}$ and its dynamic Kripke counterparts $\mathcal{F}(\mathcal{D})$. In other words, we will show that a formula is valid on a dynamic polyhedral or dynamic relational system iff it is valid on its encoding.

Lemma 5.1.11. Let $\mathcal{O} = (\mathrm{P}, \mathrm{K}, R, \mathrm{V})$ be a dynamic polyhedral or dynamic relational model. Then for any formula φ:

$$\mathcal{O} \vDash \varphi \Leftrightarrow \mathcal{M}(\mathcal{O}) \vDash \varphi.$$

Proof. $(\Rightarrow)$ Assume that $\mathcal{O} \vDash \varphi$. Take any point $\widetilde{\sigma} \in \mathcal{M}(\mathcal{O})$. Since $\widetilde{\sigma} \neq \varnothing$, we have that there is a point $z \in \widetilde{\sigma}$. Therefore, by assumption: $\mathcal{O}, z \vDash \varphi$. Hence, by Theorem 3.2.11 we have that $\mathcal{M}(\mathcal{O}), \widetilde{\sigma} \vDash \varphi$. Since $\widetilde{\sigma}$ was arbitrary, we conclude that $\mathcal{M}(\mathcal{O}) \vDash \varphi$.

$(\Leftarrow)$ Assume that $\mathcal{M}(\mathcal{O}) \vDash \varphi$. Take any point $x \in \mathrm{P}$. By assumption, we have that $\mathcal{M}(\mathcal{O}), \widetilde{\sigma}^x \vDash \varphi$. By theorem 3.2.11 it follows that $\mathcal{O}, x \vDash \varphi$. $\square$

Lemma 5.1.12. Let $\mathcal{D} = (\mathrm{P}, \mathrm{K}, R)$ be a dynamic polyhedral or dynamic relational system. The for every formula φ:

$$\mathcal{D} \vDash \varphi \Leftrightarrow \mathcal{F}(\mathcal{D}) \vDash \varphi$$

Proof. $(\Rightarrow)$ Take a valuation V on $\mathcal{F}(\mathcal{D})$. It induces a valuation V' on $\mathcal{D}$ defined as follows:

$$x \in V'(p) \text{ iff } \widetilde{\sigma}^x \in V(p)$$

Since $\widetilde{\sigma}^x$ is the smallest relative interior of a simplex such that $x \in \widetilde{\sigma}^x$, V' is well defined. Clearly, $\widetilde{\sigma} \subseteq V'(p)$ iff $\widetilde{\sigma} \in V(p)$. We obtain that for $\mathcal{O} = (\mathcal{D}, V)$, by assumption: $\mathcal{O} \vDash \varphi$. We obtain that $\mathcal{M}(\mathcal{O}) \vDash \varphi$ using the previous lemma. Thus, $\mathcal{F}(\mathcal{D}) \vDash \varphi$.

$(\Rightarrow)$ For the other direction take a valuation V on $\mathcal{D}$. Then take $\mathcal{M}(\mathcal{O})$ for $\mathcal{O} = (\mathcal{D}, \mathrm{V})$. By assumption: $\mathcal{M}(\mathcal{O}) \vDash \varphi$. Therefore, by the previous lemma $\mathcal{O} \vDash \varphi$. Since V was arbitrary, we conclude that $\mathcal{D} \vDash \varphi$. $\square$

Lemma 5.1.13. Let $\mathcal{D} = (\mathrm{P}, \mathrm{K}, R)$ be a dynamic polyhedral or dynamic relational system. Then $\mathcal{D} \vDash$ **Grz-K**

Proof. Consider encoding $\mathcal{F}(\mathcal{D}) = (\widetilde{K}, \precsim, R^*, \widetilde{V})$ of $\mathcal{D}$. Then we have that $\precsim$ is a transitive, reflexive, and Noetherian relation. Then $\mathcal{F}(\mathcal{D}) \vDash$ **grz** by Lemma 5.1.9. For the following facts, the proof is standard and can be found in [BDV01]:

1. All proposition tautologies are valid on every dynamic Kripke frame;
2. $\mathrm{K}_{\Box}$ and $\mathrm{K}_{\langle \mathrm{R} \rangle}$ are valid on dynamic Kripke frames;
3. Rules of inference $\mathrm{MP}, \mathrm{Nec}_{\Box}, \mathrm{Nec}_{\langle \mathrm{R} \rangle}, \mathrm{Uniform}$ preserve validity every dynamic Kripke frame.

Hence, we conclude that $\mathcal{F}(\mathcal{D}) \vDash$ **Grz-K**. And by Lemma 5.1.12 we obtain that $\mathcal{D} \vDash$ **Grz-**
$mathbfK$.

$\square$

Let us introduce two new formulas that describe the behaviour of relation R in $\mathcal{D} = (P, K, R)$.

- $F := \neg\langle \mathrm{R} \rangle\varphi \leftrightarrow \langle \mathrm{R} \rangle\neg\varphi$;
- $C := \Diamond\langle \mathrm{R} \rangle\varphi \rightarrow \langle \mathrm{R} \rangle\Diamond\varphi$.

Formula F is the formulation of the functionality and totality in the language of modal logic. At the same time, axiom C, as we shall see, corresponds to the property that the preimage of an element from $Sub_{\mathrm{K}}(P)$ is in $Sub_{\mathrm{K}}(P)$. In the dynamic topological logic, this axiom stands for continuity [KM07].

We will work with the following logics.

Definition 5.1.14.
- $\mathcal{DL} =$ **Grz** $\otimes$ **K** with $\mathcal{DL}$.
- $\mathcal{DRL} =$ **Grz**-**K**-C
- $\mathcal{DPL} =$ **Grz**-**K**-F-C

We call $\mathcal{DL}$ dynamic logic, $\mathcal{DPL}$ dynamic polyhedral logic, and $\mathcal{DRL}$ dynamic relational logic.

Since F is a Sahlqvist formula, we immediately have:

Lemma 5.1.15. Let $\mathcal{F}^d = (\mathrm{W}, \sqsubseteq, R)$ be a dynamic Kripke frame. Then:

$$\mathcal{F}^d \vDash F \text{ iff } R \text{ is functional.}$$

Proof. Follows from [BDV01, Theorem 4.42]. $\square$

Lemma 5.1.16. Let a dynamic Kripke frame $\mathcal{F}^d = (\mathrm{W}, \sqsubseteq, R)$, $\mathcal{F}^d \vDash$ C iff R is naively monotone.

Proof. See [BBH17, Proposition 3.7]. $\square$

Thus, we have the following lemma.

Theorem 5.1.17. For a dynamic system $\mathcal{D} = (P, K, R)$:

1. If $\mathcal{D}$ is a dynamic relational system, then $\mathcal{D} \vDash \mathcal{DRL}$;

2. If $\mathcal{D}$ is a dynamic polyhedral system, then $\mathcal{D} \vDash \mathcal{DPL}$;

Proof. By Lemma 5.1.13, we have that $\mathcal{D} \vDash \mathbf{Grz} - \mathbf{K}$ is valid on both dynamic polyhedral system and dynamic relational system.

Now, consider the case when $\mathcal{D}$ is a dynamic relational system. Then we have that for encoding $\mathcal{F}(\mathcal{D}) = (\widetilde{K}, \precsim, R^*)$ of $\mathcal{D}$ we have that $\mathcal{F}(\mathcal{D}) \vDash C$, by Lemma 3.3.12 and Lemma 5.1.16. Therefore $\mathcal{D} \vDash F$ by Lemma 5.1.12.

For the case when $\mathcal{D}$ is a dynamic polyhedral system, note that $\mathcal{D} = (P, K, f)$. Then $\mathcal{D} \vDash C$, because $\mathcal{D}$ is also a dynamic relational system. At the same time, since for encoding $\mathcal{F}(\mathcal{D}) = (\widetilde{K}, \precsim, f^*)$ of $\mathcal{D}$, f^* is a function, we have that $\mathcal{F}(\mathcal{D}) \vDash F$. Hence, by Lemma 5.1.12, we have that $\mathcal{D} \vDash F$. □

Corollary 5.1.18. $\mathcal{DPL} \subseteq \mathcal{L}(\mathbf{DPS})$ and $\mathcal{DRL} \subseteq \mathcal{L}(\mathbf{DRS})$.

Definition 5.1.19. A logic L is complete with respect to a class of dynamic systems $\mathbf{D}$, if $\mathsf{L} = \mathcal{L}(\mathbf{D})$, i.e. for each formula φ:

$$\mathsf{L} \vdash \varphi \text{ iff } \mathcal{D} \vDash \varphi, \text{ for every} \mathcal{D} \in \mathbf{D}.$$

In the next section, we will prove the right-to-left direction of implication for dynamic relational systems and dynamic polyhedral systems.

5.2 Surgery on the canonical model

To prove that $\mathcal{DRL} = \mathcal{L}(\mathbf{DRS})$ and $\mathcal{DPL} = \mathcal{L}(\mathbf{DPS})$, let us recall the well-known definition of the canonical model. The canonical modal construction is a classical approach for proving completeness. First, we define the necessary definition of maximal L-consistent sets.

Definition 5.2.1. Let Γ be a set of formulas and L be a logic. We call Γ *maximal* L-*consistent set* (L-MCS), whenever Γ is consistent and for set of formulas Γ' properly containing Γ is inconsistent.

The following lemma is well-known for the canonical model construction.

Lemma 5.2.2 (Lindenbaum's lemma)**.** Every consistent set Φ of formulas can be extended to a maximal consistent set Ψ, s.t. $\Phi \subseteq \Psi$.

Proof. See [BDV01, Lemma 4.17]. □

Since we are working with bimodal logic, we must define a canonical model with two relations: $\sqsubseteq_c$ and R_c.

Definition 5.2.3. Let $\mathsf{L} \supseteq \mathcal{DL}$. We call $\mathfrak{M}_{\mathsf{L}} = (\mathrm{W_c}, \sqsubseteq_{\mathrm{c}}, R_{\mathrm{c}}, V_{\mathrm{c}})$ a canonical model for L, if:

1. $\mathrm{W_c}$ is the set of all L-MCS;
2. $w \sqsubseteq_c v$ iff $\{\Diamond\varphi \mid \varphi \in v\} \subseteq w$;
3. $wR_{\mathrm{c}}v$ iff $\{\langle \mathrm{R} \rangle\varphi \mid \varphi \in v\} \subseteq w$;
4. $V_{\mathrm{c}}(p) = \{w \mid p \in w\}$, for all $p \in \mathrm{Prop}$.

The following two lemmas are also well-known. The proof of them is standard and can be found in Chapter 4 of [BDV01].

Lemma 5.2.4 (Existence lemma)**.** Let $\mathsf{L} \supseteq \mathcal{DL}$ and $\mathfrak{M}_{\mathsf{L}} = (W_{\mathrm{c}}, \sqsubseteq_{\mathrm{c}}, R_{\mathrm{c}}, V_{\mathrm{c}})$ be its canonical model. Then, for every $w \in \mathrm{W_c}$ and every formula φ in L, if $\Diamond\varphi \in w$ then there exists a point $v \in \mathrm{W_c}$ such that $w \sqsubseteq_{\mathrm{c}} v$ and $\varphi \in v$.

Lemma 5.2.5 (Truth lemma)**.** Let $\mathsf{L} \supseteq \mathcal{DL}$. For every $w \in \mathrm{W_c}$ and every formula φ in L,

$$\mathfrak{M}_{\mathsf{L}}, w \models \varphi \text{ iff } \varphi \in w.$$

Lemma 5.2.6. Let $\mathsf{L} \supseteq \mathcal{DL}$ and $\mathfrak{M}_{\mathsf{L}} = (W_{\mathrm{c}}, \sqsubseteq_{\mathrm{c}}, R_{\mathrm{c}}, V_{\mathrm{c}})$ be its canonical model. Then the following hols:

1. $\sqsubseteq_c$ is transitive and reflexive.
2. If $\mathrm{C} \in \mathsf{L}$, then R_c is naively monotone.
3. If $\mathrm{F} \in \mathsf{L}$, then R_c is a function

Proof. For the proof of item 1 see [Cha97, Proposition 5.48]. For items $2, 3$, we use the Sahlqvist theorem [BDV01, Theorem 4.42]. $\square$

We will construct a finite relation on the canonical model for every logic $\mathsf{L} \supseteq \mathcal{DL}$. This relation will depend on a finite set Φ of $\mathcal{L}_{\Box,\langle \mathrm{R} \rangle}$-formulas on W_c.

Remark 5.2.7. The idea of constructing this finitary relation was developed by David Fenández-Duque and Yoàv Montacute in [FM21]. In this paper, the authors create a foundation for a general proof method for the finite model property of bimodal logics which contain F and C as axioms. In particular, they prove that logic GL-K-F-C (they denote as **GLC**) has finite Kripke model property. We use the same approach with adjustments to the specifics of our cases to obtain finite model property for $\mathcal{DPL}$ and $\mathcal{DRL}$.

The method we use combines selective filtration from [Cha97] and the method of φ-final sets from [FM21].

For further discussion fix a logic $\mathsf{L} \supseteq \mathcal{DL}$ and its canonical model $\mathfrak{M}_{\mathsf{L}} = (W_c, \sqsubseteq_c, R_c, V_c)$.

For this purpose, we will need the following definition:

Definition 5.2.8. A point $w \in \mathrm{W}_c$ is said to be a φ-final point if $\varphi \in w$, and whenever $w \sqsubseteq_c v$ and $\varphi \in v$, it follows that $v \sqsubseteq_c w$.

We also recall here Zorn's since we will need it in our proof.

Lemma 5.2.9 (Zorn's lemma)**.** Let $(A, \leq)$ be a preordered set where A is nonempty. Suppose that every chain $\mathcal{C}$ has an upper bound in A. Then, A has a $\leq -$ maximal element.

Proof. See [Jec03, Theorem 5.4]. $\square$

To construct the relation mentioned above, we need the following two lemmas.

Lemma 5.2.10. Let w be a point in W_c and $\Diamond\varphi \in w$. Then there exists some φ-final point v such that $w \sqsubseteq_c v$.

Proof. The same lemma was proved in [BBF21, Lemma 6.6].

Let $\Diamond\varphi \in w \in \mathrm{W}_c$. Take $S = \{v \mid w \sqsubseteq_c v, \varphi \in v\}$. We will show that every chain of elements in S has a supremum.

Take a chain C in S. Define $\Phi = \{\varphi\} \cup \{\Box\varphi \mid \Box\varphi \in v, v \in C\}$. Showing that Φ is consistent and using Lindenbaum's lemma, we will obtain a maximally consistent set of formulas, which will be a point in our canonical model.

Suppose that Φ is inconsistent. Then there are elements $\Box\varphi_1 \in v_1, \ldots, \Box\varphi_n \in v_n$ with $\{v_1, \ldots, v_n\} \subseteq C$ such that the set $\{\varphi, \Box\varphi_1, \ldots, \Box\varphi_n\}$ is inconsistent. Without loss of generality we can assume that $v_1 \sqsubseteq_c \ldots \sqsubseteq_c v_n$. By transitivity, we obtain that $\{\Box\varphi_1, \ldots, \Box\varphi_n\} \subseteq v_n$. Since $v_n \in C \subseteq S$, we also have $\varphi \in v_n$. This contradicts that v_n is consistent.

Apply Lindenbaum's lemma to Φ to obtain $u \in \mathrm{W}_c$ such that $\Phi \subseteq u$. Since $\sqsubseteq_c$ is reflexive, for every $\Box\varphi \in \Phi$ we have that $\varphi \in u$. Hence, by definition of canonical relation, $v \sqsubseteq_c u$ for every $v \in C$. Thus, u is an upper bound of C and $u \in C$.

Applying Zorn's lemma, we obtain an element $u' \in S$. So, for every $t \in S$ we have that $t \sqsubseteq_c u$ by the maximality of u. Transitivity of $\sqsubseteq_c$ implies that u' is φ-final. $\square$

Lemma 5.2.11. Let $w \in \mathrm{W}_c$ and φ be a formula. We have that $\Diamond\varphi \in w$ iff there exists some φ-final point v such that $w \sqsubseteq_c v$ and $\mathfrak{M}_c, v \vDash \Box(\neg\varphi \to \Box\neg\varphi)$.

Proof. ($\Rightarrow$) By Lemma 5.2.10 there is a φ-final point u, s.t. $w \sqsubseteq_c u$. By the truth lemma we have that $\mathfrak{M}_{\mathsf{L}}, u \vDash \mathbf{grz}$. It is possible to rewrite $\mathbf{grz}$ as $\varphi \to \Diamond(\Box(\neg\varphi \to \Box\neg\varphi) \wedge \varphi)$. Since u is φ-final we must have that $\mathfrak{M}_c, u \vDash \Diamond(\Box(\neg\varphi \to \Box\neg\varphi) \wedge \varphi)$. By the semantics, we have that there is a successor v of u, s.t. $\mathfrak{M}_c, v \vDash \varphi$ and $v \vDash \Box(\neg\varphi \to \Box\neg\varphi)$. The fact that $v \vDash \varphi$ and $u \sqsubseteq_c v$ imply that v is φ-final as well.

($\Leftarrow$) The result follows by the definition of canonical relation $\sqsubseteq_c$. $\Box$

Definition 5.2.12. Let $\mathcal{F} = (W, \sqsubseteq)$ be a Kripke frame. We call $C \subseteq W$ a *cluster*, if for every two points $v_1, v_2 \in C$: $v_1 \sqsubseteq v_2 \sqsubseteq v_1$.

For every point v, let us denote the cluster of v with $C(v)$.

For every cluster C of points in W define

$$\sqsubseteq (C) = \bigcup\{\sqsubseteq (v) \mid v \in C\}.$$

Now we are fully equipped to define the finitary relation that we need. The following lemma is the adaptation of [FM21, Lemma 5.5], for the case of $\mathbf{GL}$.

Lemma 5.2.13. Let Φ be a finite set of formulas closed under subformulas. There is an auxiliary relation $\sqsubseteq_\Phi$ on the canonical model of L such that:

1. If $x \sqsubseteq_\Phi y$, then $\sqsubseteq_c$;
2. For each $w \in \mathrm{W}_c$, the set $\sqsubseteq_\Phi (w)$ is finite;
3. If $\Diamond\varphi \in w \cap \Phi$, then there exists $v \in \mathrm{W}$ with $w \sqsubseteq_\Phi v$ and $\varphi \in v$;
4. $\sqsubseteq_\Phi$ is reflexive and transitive and Noetherian.

Proof. Using lemma 5.2.11 and the axiom of choice, we can obtain a function that for each formula φ and cluster C with $\Diamond\varphi \in \bigcup C$, assigns a φ-final point $w(\varphi, C)$ such that $w(\varphi, C) \in \sqsubseteq_c (C)$ and $\mathfrak{M}_c, w(\varphi, C) \vDash \Box(\neg\varphi \to \Box\neg\varphi)$.

Set $u \sqsubseteq^0_\Phi v$ iff $u \sqsubseteq_c v$ and there exists $\varphi \in \Phi$ such that $\Diamond\varphi \in u$, $v = w(\varphi, C(u))$ and $\mathfrak{M}_c, u \nvDash \varphi$.

Observe that $u \sqsubseteq^0_\Phi v$ implies that $\neg(v \sqsubseteq_c u)$. Indeed, if $v \sqsubseteq_c u$, then $\mathfrak{M}_c, u \vDash \Box\neg\varphi$. But this contradicts φ-finality of v. This also entails that $\sqsubseteq^0_\Phi$ is irreflexive.

Take $\sqsubseteq_\Phi$ to be a transtivie and reflexive closure of $\sqsubseteq^0_\Phi$. Conditions $1, 3$ follow instantly from the construction. Let us show that condition 2 holds as well.

First, observe that $\sqsubseteq^0_\Phi(x)$ is finite for every x, since $|\Phi|$ is finite and every point can have at most $|\Phi|$ different successors. Define $\sqsubset_\Phi = \sqsubseteq_\Phi \setminus \{(w, w) \mid w \in \mathrm{W}_c\}$. Clearly, if we show that $\sqsubset_\Phi (u)$ is finite, so will be $\sqsubseteq_\Phi (u)$. We will take the unravelling of $\sqsubset_\Phi (u)$ to prove by contradiction that it is finite. For every $v \sqsupset_\Phi u$ there is a sequence:

$$u \sqsubset^0_\Phi v_1 \sqsubseteq^0_\Phi \dots \sqsubset^0_\Phi v_n = v.$$

By taking the minimal sequence, we may assume that the map from points to sequences is injective. Consider the tree, which consists of all such paths, ordered by the initial segment relation. This is a finitely branching tree, since $\sqsubseteq^0_\Phi(x)$ is always finite for every x. Assume that $\sqsubset_\Phi (u)$ is infinite. Then the tree is infinite. Moreover, by Kőnig's lemma, there is an infinite sequence

$$u \sqsubset^0_\Phi v_1 \sqsubset^0_\Phi \dots.$$

By definition of $\sqsubseteq^0_\Phi$ for each $i \in \omega$ there is $\varphi_i \in \Phi$ such that v_{i+1} is φ_i-final and $\mathfrak{M}_c, v_{i+1} \vDash \Box(\neg\varphi \to \Box\neg\varphi)$ and $\mathfrak{M}_c, v_i \nvDash \varphi$. Since $\sqsubseteq^0_\Phi$ is a subset of $\sqsubseteq_c$, we have that $v_i \sqsubseteq_c v_j$ whenever $i \leq j$. Since Φ is finite, there is some $\delta \in \Phi$ such that v_i if δ-final for infinitely many values of i. Let i_0 be the least such value. If $i' > i_0$ is any other such value for $v_{i'}$ which is δ-final, we have that $v_{i_0} \sqsubseteq^0_\Phi u_1 \sqsubseteq^0_\Phi \dots \sqsubseteq^0_\Phi v_{i'}$. Then by transitivity of $\sqsubseteq_\text{c}$ we have that $v_{i_0} \sqsubseteq_c u_1 \sqsubseteq_c v_{i'}$. Since $v_{i'}$ and v_{i_0} are δ-final and $v_{i_0} \sqsubseteq_\text{c} v_{i'}$, then they are in the same cluster. But then that $u_1 \in C(v_{i_0})$. This contradicts the fact that $\neg(u_1 \sqsubseteq_c v_{i_0})$.

A similar strategy is applied to show that $\sqsubseteq_\Phi$ has no cycles. Assume that there is a cycle $u \sqsubseteq_\Phi v_1 \sqsubseteq_\Phi v_2 \sqsubseteq_\Phi \dots \sqsubseteq_\Phi v_n = u$, s.t. $v_i \neq v_j$ for every $i \neq j$. Since $\sqsubseteq_\Phi$ is a transitive closure of $\sqsubseteq^0_\Phi$, we have that $u \sqsubseteq^0_\Phi v_0' \dots \sqsubseteq^0_\Phi v_n = u$. Since we have that $\sqsubseteq_\Phi \subseteq \sqsubseteq_c$, we obtain that $v_0' \in C(u)$. This contradicts the fact that $\neg(v_0' \sqsubseteq_c u)$. $\Box$

Lastly, we will define finitary relation R_Φ, also using the finite set of formulas Φ. This relation will be essential for constructing a finite model. Having a Φ-finite set of formulas closed under subformulas, and element $w \in W_c$, assume that $\langle\text{R}\rangle\varphi \in w \cap \Phi$. Then there is an element w' such that wR_cw' and $\varphi \in w'$ by Lemma 5.2.4. Hence, using the choice function, choose one such w' for each $\langle\text{R}\rangle\varphi \in w \cap \Phi$. Denote the set of these chosen elements with $X_w = \{x(w, \langle\text{R}\rangle\varphi) \mid \langle\text{R}\rangle\varphi \in w \cap \Phi\}$.

Define $R_\Phi \subseteq R_c$ in the following way:

$$wR_\Phi y \text{ iff } y \in X_w. \tag{5.1}$$

Observe that in the case when R_c is a function, we do not need to use the axiom of choice since there is only one element v for every $w \in W_c$ such that wR_cv.

5.3 Story about moments in a forest with a Φ-morphism

We now describe the construction used to prove the completeness result with respect to classes of finite models for our dynamic logics $\mathcal{DPL}$ and $\mathcal{DRL}$. The key definitions and techniques are taken from [FM21].

Definition 5.3.1. We will say that a Kripke frame $\mathcal{F} = (W, \sqsubseteq)$ is a *tree*, if whenever $a \sqsubseteq c$ and $b \sqsubseteq c$, it follows that $a \sqsubseteq b$ or $b \sqsubseteq a$.

Definition 5.3.2. Let $\mathcal{F} = (W, \sqsubseteq)$ be a Kripke frame. If there is y such that for every x: $\sqsubseteq_{\mathfrak{m}}$ $y \sqsubseteq x$, then we call y a root of $\mathcal{F}$.

Definition 5.3.3. Let $\mathcal{F} = (W, \sqsubseteq)$ be a Kripke frame. We say that y is a *strict* $\sqsubseteq$ *successor* of x, if $x \sqsubseteq y$ and $\neg(y \sqsubseteq x)$.

Definition 5.3.4. Let $\mathcal{F} = (W, \sqsubseteq)$ be a Kripke frame. We say that y is an *immediate* $\sqsubseteq$ *successor* of x if there is no $z \neq y$ such that $x \sqsubseteq z \sqsubseteq y$.

Definition 5.3.5. Structure $\mathfrak{m} = (|\mathfrak{m}|, \sqsubseteq_{\mathfrak{m}}, \nu_{\mathfrak{m}}, r_{\mathfrak{m}})$ we call a moment whenever $(|\mathfrak{m}|, \sqsubseteq_{\mathfrak{m}})$ is finite Noetherian tree, with a root $r_{\mathfrak{m}}$, and $\nu_{\mathfrak{m}}$ is a valuation on $\mathfrak{m}$.

Definition 5.3.6 (Forest)**.** Assume that $(\mathfrak{m}_i)_{i<I}$ are moments for a set of indexes I. Then their *forest* is the structure $\mathfrak{F} = (|\mathfrak{F}|, \sqsubseteq, \nu, C)$, where:

- $|\mathfrak{F}| = \bigsqcup\limits_{i<I} |\mathfrak{m}_i|$;
- $\sqsubseteq = \bigsqcup\limits_{i<I} \sqsubseteq_i$;
- $\nu(p) = \bigsqcup\limits_{i<I} \nu_i$;
- $C = \{r_{\mathfrak{m}_i}\}_{i<I}$.

Definition 5.3.7. A *story* (with duration I) is a tuple $\mathbf{S} = (|\mathbf{S}|, \sqsubseteq_{\mathbf{S}}, R_{\mathbf{S}}, \nu_{\mathbf{S}}, C_{\mathbf{S}})$ such that there are forests $\mathfrak{F}_i = (|\mathfrak{F}_i|, \sqsubseteq_i, \nu_i, C_i)$ for each $i < I$, and relations $(R_i)_{i<I}$ such that:

- $|\mathbf{S}| = \bigsqcup\limits_{i<I} |\mathfrak{F}_i|$;
- $\sqsubseteq_{\mathbf{S}} = \bigsqcup\limits_{i<I} \sqsubseteq_i$;
- $\nu_{\mathbf{S}}(p) = \bigsqcup\limits_{i<I} \nu_i$;

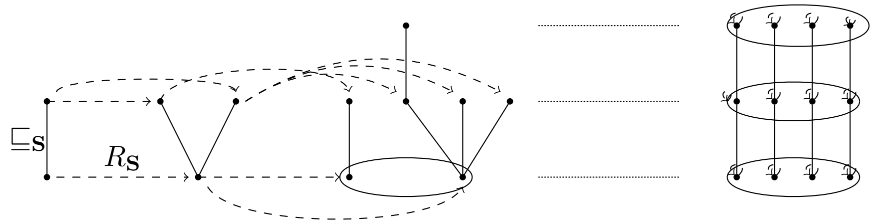

Figure 5.1: An example of a Story. The straight arrows represent the relation $\sqsubseteq_{\mathbf{S}}$ while dashed arrows represent the dynamic relation $R_{\mathbf{S}}$. Each vertical slice represents a moment.

- $R_{\mathbf{S}} = Id_I \cup \bigsqcup_{i<I} R_i$, where $R_i : |\mathfrak{F}_i| \times |\mathfrak{F}_{i+1}|$ is a naively monotone relation such that $R_i(C_i) = C_{i+1}$, and R_I is the identity on $|\mathfrak{F}_I|$.
- $C_{\mathbf{S}} = C_0$.

Observe that every story is a naively monotone dynamic model.

Recall that a p-morphism between Kripke models is a type of map that preserves validity. It can be defined in the context of dynamic Kripke frames as follows:

Definition 5.3.8. Let $\mathcal{F}_1^{\mathcal{D}} = (W_1, \sqsubseteq_1, R_1)$ and $\mathcal{F}_2^{\mathcal{D}} = (W_2, \sqsubseteq_2, R_2)$ be two dynamic Kripke frames. We say that $f : W_1 \to W_2$ is a dynamic p-morphism, whenever:

1. $x \sqsubseteq_1 y$ implies that $f(x) \sqsubseteq_2 f(x)$;
2. $f(x) \sqsubseteq_2 z$ implies that $\exists y : x \sqsubseteq_2 y$ and $f(y) = z$;
3. xR_1y implies that $f(x)R_2f(x)$;
4. $f(x)R_2z$ implies that $\exists y : xR_1y$ and $f(y) = z$.

If f satisfies only $1, 2$, we call f a p-morphism.

If there is a surjective dynamic p-morphism between $\mathcal{F}_1$ and $\mathcal{F}_2$, then we will denote this by $\mathcal{F}_1 \twoheadrightarrow \mathcal{F}_2$.

We will use the following fact about dynamic morphism:

Lemma 5.3.9. Let $\mathcal{F}_1^{\mathcal{D}}$ and $\mathcal{F}_2^{\mathcal{D}}$ be dynamic Kripke frames. Assume that $\mathcal{F}_1 \twoheadrightarrow \mathcal{F}_2$. Then $\mathcal{F}_2^{\mathcal{D}} \vDash \varphi$, whenever $\mathcal{F}_1^{\mathcal{D}} \vDash \varphi$.

Proof. See [BDV01, Theorem 3.14].

□

Definition 5.3.10 (Φ-morphism)**.** Let $\mathsf{L} \supset \mathcal{DL}$ be a logic, $\mathfrak{M}_{\mathsf{L}} = (W_{\mathrm{c}}, \sqsubseteq_{\mathrm{c}}, R_{\mathrm{c}}, V_{\mathrm{c}})$ its canonical model, and $\mathbf{S} = (|\mathbf{S}|, \sqsubseteq_{\mathbf{S}}, R_{\mathbf{S}}, \nu_{\mathbf{S}}, C_{\mathbf{S}})$ be a story of duration I. A map $\pi : |\mathbf{S}| \to W_{\mathrm{c}}$ is called a dynamic Φ-morphism if for all $x \in |\mathbf{S}|$ the following conditions are satisfied:

1. $x \in \nu_{\mathbf{S}}(p) \Longleftrightarrow p \in \pi(x)$;
2. If $x \in |\mathfrak{F}_i|$ for some $i < I$, then $R_{\Phi}(\pi(x)) \subseteq \pi(R_{\mathbf{S}}(x)) \subseteq R_{\mathrm{c}}(\pi(x))$;
3. If $x \sqsubseteq_{\mathbf{S}} y$ then $\pi(x) \sqsubseteq_c \pi(y)$;
4. If $\pi(x) \sqsubseteq_{\Phi} v$ for some $v \in W_{\mathrm{c}}$, then there exists $y \in \mathbf{W}$ such that $x \sqsubseteq_{\mathbf{S}} y$ and $v = \pi(y)$;

If π satisfies all the conditions except for 2, we will say that π is a Φ-morphism.

Remark 5.3.11. In case $F \in \mathsf{L}$, we have that R_c in the canonical model becomes a function. Thus, the second condition of the previous definition becomes $R_{\Phi}(\pi(x)) = \pi(R_{\mathbf{S}}(x))$ and $\pi(R_{\mathbf{S}}(x)) = R_c(\pi(x))$ (in case $R_{\Phi}(\pi(x)) \neq \varnothing$).

We define the depth of the formula φ as follows:

Definition 5.3.12. For a formula φ in $\mathcal{L}_{\Box,\langle\mathrm{R}\rangle}$, $\mathbf{depth}(\varphi)$ is defined recursively:

1. $\mathbf{depth}(p) = 0$, if $p \in \mathrm{Prop}$;
2. $\mathbf{depth}(\neg\varphi) = \mathbf{depth}(\varphi)$;
3. $\mathbf{depth}(\varphi \wedge \psi) = max\{\mathbf{depth}(\varphi); \mathbf{depth}(\psi)\}$;
4. $\mathbf{depth}(\langle\mathrm{R}\rangle\varphi) = \mathbf{depth}(\varphi) + 1$.

Lemma 5.3.13. Let $\mathsf{L} \supseteq \mathcal{DL}$ be a logic and $\pi : \mathbf{S} \to W_c$ be a dynamic Φ-morphism L-story $\mathbf{S}$ of duration I to the canonical model $\mathfrak{M}_{\mathsf{L}}$. Then for every formula $\varphi \in \Phi$ of $\langle\mathrm{R}\rangle$-depth at most I:

$$\varphi \in \pi(x) \text{ iff } x \in [\![\varphi]\!]^{\mathbf{S}}, \text{ for every } x \in |\mathfrak{F}_0|$$

Proof. We prove the more general claim that if φ has $\langle\mathrm{R}\rangle$-depth at most $I - j$ and $x \in |\mathfrak{F}_j|$, then $\varphi \in \pi(x)$ iff $x \in [\![\varphi]\!]^{\mathbf{S}}$. We proceed by induction on the complexity of φ. The case of propositional variables follows by definition of Φ-morphism. The case of boolean connectives follows by standard arguments.

Consider the case of $\Box\varphi$. Assume that $\Box\varphi \in \pi(x)$, and $x \sqsubseteq_{\mathbf{S}} y$ for some y. Then $y \in |\mathfrak{F}_j|$, since $\mathfrak{F}_j$ is closed under $\sqsubseteq_{\mathbf{S}}$. Then $\pi(x) \sqsubseteq_c \pi(y)$ and $\varphi \in \pi(y)$ by Lemma 5.2.5. Hence, $y \in [\![\varphi]\!]^{\mathbf{S}}$ by the inductive hypothesis. Since y was arbitrary, we conclude that $x \in [\![\varphi]\!]^{\mathbf{S}}$. For the other direction, assume that $x \in [\![\Box\varphi]\!]^{\mathbf{S}}$. To show that $\Box\varphi \in \pi(x)$ it is sufficient to prove that $\mathfrak{M}_c, \pi(x) \vDash \Box\varphi$. Take arbitrary z, s.t. $\pi(x) \sqsubseteq_c z$. Then there exists $y \in \mathbf{S}$ such that $x \sqsubseteq_{\mathbf{S}} y$ and $z = \pi(y)$. Therefore $y \in [\![\varphi]\!]^{\mathbf{S}}$ and by the fact that $y \in |\mathfrak{F}_j|$ and induction hypothesis, we have that $\varphi \in \pi(y)$. But this means that $\mathfrak{M}_c, \pi(y) \vDash \varphi$. We conclude that $\mathfrak{M}_c, \pi(x) \vDash \Box\varphi$.

Let us consider the case of $\langle\mathrm{R}\rangle\varphi$. If $\langle\mathrm{R}\rangle\varphi \in \pi(x)$, then there is y such that $\pi(x) R_\Phi y$, since $\langle\mathrm{R}\rangle\varphi \in \Phi$. By definition of Φ-morphism, we have that there is a u such that $x R_{\mathbf{S}} u$ and $\pi(u) = y$. Then we have that $u \in |\mathfrak{F}_{j+1}|$, but at the same time the depth of φ is at most $I - j - 1$. Therefore, $u \in [\![\varphi]\!]^{\mathbf{S}}$. Hence, by definition $x \in [\![\langle\mathrm{R}\rangle\varphi]\!]^{\mathbf{S}}$.

If $x \in [\![\langle\mathrm{R}\rangle\varphi]\!]^{\mathbf{S}}$, then there is y, s.t. $x R_{\mathbf{S}} y$ and $y \in [\![\varphi]\!]^{\mathbf{S}}$. By definition of dynamic Φ-morphism, we have that $\pi(x) R_c \pi(y)$. Again, applying the inductive hypothesis as above, we have that $\varphi \in \pi(y)$ and therefore $\langle\mathrm{R}\rangle\varphi \in \pi(x)$. $\Box$

We devote the rest of the section to constructing the right story and a Φ-morphism on it.

Definition 5.3.14. Let $\mathsf{L} \supseteq \mathcal{DL}$, and $\mathfrak{M}_{\mathsf{L}} = (W_c, \sqsubseteq_c, R_c, V_c)$ be its canonical model. For a sequence of moments $(\mathfrak{a}_m)_{m<N}$ and $x \in W_c$ in W_c. Define $\mathfrak{n} = \binom{\vec{\mathfrak{a}}}{x}$ as follows:

1. $|\mathfrak{n}| = \{x\} \bigsqcup \bigsqcup_{m<N} |\mathfrak{a}_m|$;
2. $y \sqsubseteq_{\mathfrak{n}} z$ if:
   - $y = x$ and $z \in |\mathfrak{a}_m|$ for some $\mathfrak{a}_m$ or $z = x$;
   - $y, z \in |\mathfrak{a}_m|$ and $y \sqsubseteq_m z$ for some $\mathfrak{a}_m$;

3. $\nu_{\mathfrak{n}}(p) = V_c \restriction_{\{x\}} (p) \bigsqcup \bigsqcup_{m<N} \nu_{\mathfrak{a}_m}(p)$;

4. $r_{\mathfrak{n}} = x$.

Lemma 5.3.15. If $(\mathfrak{a}_m)_{m<N}$ is a sequence of moments, and $x \in W_c$, then $\mathfrak{n} = \binom{\vec{\mathfrak{a}}}{x}$ is a moment.

Proof. Every $\mathfrak{a}_m$ is a Noetherian tree, which is why following Definition 5.3.14, we have to check that the addition of x does not break anything in our relation. Reflexivity and transitivity of the relation follow from the definition. To observe that $\sqsubseteq_{\mathfrak{n}}$ is Noetherian, note that $\forall z \in |\mathfrak{n}| : x \sqsubseteq z$, but not vise versa. Also, it follows that x is a root. $\square$

Lemma 5.3.16. Let $\mathsf{L} \supseteq \mathcal{DL}$, and $\mathfrak{M}_{\mathsf{L}}$ be its canonical model. Then for each $w \in W_c$ there exists a moment $\mathfrak{m}$ and a Φ-morphism $\pi : W \to W_{\mathrm{c}}$ such that $\pi(r_{\mathfrak{m}}) = w$.

Proof. We prove that there is a moment $\mathfrak{m}$ and a map $\pi : |\mathfrak{m}| \to W_c$ that is a p-morphism on the structure $(W_c, \sqsubseteq_\Phi)$.

First we define strict successor-relation $\sqsubseteq^1_\Phi$ such that if $x \sqsubseteq^1_\Phi y$, then $x \sqsubseteq_\Phi y$. This relation is converse well-founded, so we can prove the statement by induction on it. Thus, for all v such that $w \sqsubseteq^1_\Phi v$ there is a moment $\mathfrak{m}_v$ and a p-morphism $\rho_v : |\mathfrak{m}_v| \to W_c$ with respect to $\sqsubseteq_\Phi$ that maps the root of $r_{\mathfrak{m}_v}$ to v. Take the sequence of moments $\vec{\mathfrak{m}} = (\mathfrak{m}_v)_v$. Then define $\mathfrak{m} = \binom{\vec{\mathfrak{m}}}{w}$. The corresponding Φ-morphism $\pi : |\mathfrak{m}| \to W_c$ is constructed in a natural way:

$$\pi(x) = \begin{cases} w, & \text{if } x = w; \\ \rho_v(x), & \text{if } x \in |\mathfrak{m}_v| \end{cases}$$

Clearly, π is a p-morphism. $\square$

Now we will prove that we can extend our stories.

Lemma 5.3.17. Let $\mathsf{L} \supseteq \mathcal{DRL}$ and $\mathfrak{M}_{\mathsf{L}} = (W_{\mathrm{c}}, \sqsubseteq_{\mathrm{c}}, R_{\mathrm{c}}, \nu_{\mathrm{c}})$ be its canonical model. Suppose that there exists a Φ-morphism $\pi : |\mathfrak{m}| \to W_c$ such that $R_\Phi(\pi(|\mathfrak{m}|)) \neq \varnothing$. Then there exist sequences of

- pairwise disjoint moments $(t_i)_{i<N}$;
- naively monotone relations $(R_i)_{i<N}$, s.t. $R_i \subseteq |\mathfrak{m}| \times |t_i|$ with $R_i(r_{\mathfrak{m}}) = r_{t_i}$;
- Φ-morphisms $\rho_i : |t_i| \to W_{\mathrm{c}}$;

such that:

- $\rho = \bigsqcup_{i<N} \rho_i$ is a Φ-morphism from forest $\mathfrak{F}$ generated by $(t_i)_{i<N}$;

- $R = \bigsqcup_{i<N} R_i$ is a naively monotone map such that $R_\Phi(\pi(w)) \subseteq \rho(R(w))$ and $\rho(R(w)) \subseteq R_c(\pi(w))$ for every $w \in |\mathfrak{m}|$.

Proof. We proceed by the induction on height of $\mathfrak{m}$.

Assume that for every moment $\mathfrak{n}$, s.t. **height**$(\mathfrak{n})$ $<$ **height**$(\mathfrak{m})$ the statement holds. Take the root r of the moment $\mathfrak{m}$. Our proof will consist of three parts. The first part will ensure that $R_\Phi(\pi(w)) \subseteq \rho(R(w))$, the second part will show that $\rho(R(w)) \subseteq R_c(\pi(w))$, and the third part combines these two conditions.

$\mathbf{Part 1}$:

Take $R_\Phi(\pi(r)) = \{x_1, \ldots, x_n\}$. Then for every x_i by Lemma 5.3.16 there is a moment $\mathfrak{x}_i$ and Φ-morphism $\delta_i : |\mathfrak{x}_i| \to W_c$ such that $\delta_i(r_{\mathfrak{x}_i}) = x_i$. In case some of the moments intersect for $i < j$, take an isomorphic copy of $\mathfrak{x}_i$, and adjust the δ_i. Define $\rho' = \bigsqcup_{i \leq n} \delta_i$. Then $\rho' : \bigsqcup_{i \leq n} \mathfrak{x}_i \to W_c$ is a Φ-morphism. Define also $R_i \subseteq |\mathfrak{m}| \times |\mathfrak{x}_i|$ in the following way:

$$wR_iv \text{ iff } w = r \text{ and } v = r_{\mathfrak{x}_i}.$$

Take $R' = \bigcup_{i \leq n} R_i$.

Then clearly R' is naively monotone and since $R_\Phi(\pi(r)) = \rho'(R'(r))$ we have that $\rho'(R'(r)) \subseteq R_c(\pi(r))$.

$\mathbf{Part 2}$:

Take $\mathcal{A} = \{a_1, \ldots, a_m\}$ the set of all strict successors of r. For each a_i we can take the generated by a_i submodel α_i. Then α_i is a moment, s.t. **height**$(\alpha_i) <$ **height**$(\mathfrak{m})$. We will consider only such a_i, for which $R_\Phi(\pi(\alpha_i)) \neq \varnothing$. By I.H. we have that for each such α_i there are:

- pairwise disjoint moments $(\mathfrak{m}^i_j)_{j<N_i}$;

- naively monotone relations $(R^i_j)_{i<N_i}$, s.t. $R^i_j \subseteq |\alpha_i| \times |\mathfrak{m}^i_j|$, with $R^i_j(r_{\alpha_i}) = r_{\mathfrak{m}^i_j}$;

- Φ-morphisms $\rho^i_j : |\mathfrak{m}^i_j| \to W_\mathrm{c}$;

such that:

- $\rho^i = \bigsqcup_{i<N} \rho^i_j$ is a Φ-morphism from forest $\mathfrak{F}_i$ generated by $(\mathfrak{m}^i_j)_{j<N_i}$;

- $R_i = \bigsqcup_{i<N} R^i_j$ is a naively monotone map such that $R_\Phi(\pi(w))$ and $R_\Phi(\pi(w)) \subseteq \rho^i(R(w)) \subseteq R_c(\pi(w))$ for every $w \in |\alpha_i|$.

Fix some i. Take the sequence of moments $(\mathfrak{m}^i_j)_{j<N_i}$. Fix a moment $\mathfrak{m}^i_j$ for some j. Take the root c of $\mathfrak{m}^i_j$. Then $c = R^i_j(a_i)$, and therefore $c \in R^i(a_i)$. Hence, $\rho^i(c) \in \rho(R^i(a_i))$. Thus, $\rho^i(c) \in R_c(\pi(a_i))$. By monotonicity of π we obtain that $\pi(r) \sqsubseteq_c \pi(a_i)$. Since $\pi(a_i)R_c\rho^i(c)$, we have that $\exists c' : \pi(r)R_c c' \sqsubseteq_c \rho^i(c)$, because R_c is naively monotone. Take one such c'. Take the set $C = \{c_1, \ldots, c_k\}$ of all immediate strict $\sqsubseteq_\Phi$ successors of c'. Then for all $c_l \in C$ there is a moment $\mathfrak{c}_l$ and Φ-morphism $\varepsilon_l : |\mathfrak{c}_l| \to W_c$ with $\varepsilon_l(r_{\mathfrak{c}_l}) = c_l$. Take $\vec{\mathfrak{c}} = (\mathfrak{c}_l)_{l\leq k}$. Define $\mathfrak{n}^i_j = \binom{\vec{\mathfrak{c}} * \mathfrak{m}^j_i}{c'}$, and define $P^i_j \subseteq |\mathfrak{m}| \times \mathfrak{n}^i_j$ in the following way:

$$wP^i_jv \text{ iff } wR^i_jv \text{ or } (w = r \text{ and } v = c').$$

Then P^i_j is naively monotone and $P^i_j(r) = r_{\mathfrak{n}^i_j}$.

Define $\theta^i_j : |\mathfrak{n}^i_j| \to W_c$:

$$\theta^i_j(w) = \begin{cases} \rho^i_j(w), & \text{if } w \in |\mathfrak{m}^i_j|; \\ c', & \text{if } w = c'; \\ \varepsilon_l(w), & \text{if } w \in |\mathfrak{c}_l|. \end{cases}$$

It is easy to see that θ^i_j is a Φ-morphism.

Now we can take the sequence of moment $(\mathfrak{n}^i_j)_{j<N_i}$. If $t^i_{j_1} \cap t^i_{j_2} \neq \varnothing$ for some $j_1 < j_2$, then take an isomorphic copy j_1, and adjust the corresponding $\theta^i_{j_1}$ and $P^i_{j_1}$.

Now take $\theta^i = \bigsqcup_{j<N_i} \theta^i_j$ and $P^i = \bigsqcup_j P^i_j$. Since $\theta^i_j \upharpoonright_{\mathfrak{m}^i_j} = \rho^i_j$ and $P^i_j \upharpoonright_{\mathfrak{m}^i_j} = P^i_j$, we conclude that $R_\Phi(\pi(x)) \subseteq \theta^i(P^i(x))$ and $\theta^i(P^i(x)) \subseteq R_c(\pi(x))$ for all $x \in \alpha_i$.

Thus, to check that $\theta^i(P^i(x)) \subseteq R_c(\pi(x))$ for all $x \in |\mathfrak{m}|$, observe that when $x = r$, we have that for every $c' \in P^i(r) : \theta^i(c') = c'$ and $\pi(x)R_cc'$.

Part3 :

Finalizing the proof, define $R = R' \cup \bigsqcup_{i\leq m} P_i$, and $\rho = \rho' \cup \bigsqcup_{i\leq m} \theta^i$. Observe that:

1. R' is a naively monotone relation on pairwise disjoint sets;
2. P_i for every $i \leq m$ is a naively monotone relation pairwise disjoint sets;
3. ρ' is a Φ-morphism;

4. θ^i for all $i \leq m$ is a Φ-morphisms.

We conclude that R is naively monotone, ρ is a Φ-morphism such that $R_\Phi(\pi(x)) \subseteq \rho(R(x))$ and $\rho(R(x)) \subseteq R_c(\pi(x))$ for all $x \in |\mathfrak{m}|$. $\square$

Proposition 5.3.18. Let $\mathsf{L} \supseteq \mathcal{DL}$ be a logic and $\mathfrak{M}_\mathsf{L} = (W_\mathrm{c}, \sqsubseteq_\mathrm{c}, g_\mathrm{c}, \nu_\mathrm{c})$ be its canonical model. Assume that $I < \omega$ and $w \in W_\mathrm{c}$. Then there is a story $\mathbf{S}$ of duration I and a dynamic Φ-morphism $\pi : |\mathbf{S}| \to W_c$ with $w = \pi(C_\mathbf{S})$ and $|C_\mathbf{S}| = 1$.

Proof. We prove the statement by induction on I. In case $I = 0$, we just use Lemma 5.3.16. For the induction step assume that there is a story $\mathbf{S}'$ of length I and a dynamic Φ-morphism $\pi' : |\mathbf{S}'| \to W_c$. Take the forest $\mathfrak{F}_I$. Then in consists of moments $\mathfrak{m}_1, \ldots, \mathfrak{m}_n$. Divide these moments into two parts:

1. $A_1 = \{\mathfrak{v}_1, \ldots, \mathfrak{v}_{k_1}\}$ such that $R_\Phi(\pi(\mathfrak{v}_i)) \neq \varnothing$
2. $A_2 = \{\mathfrak{b}_1, \ldots, \mathfrak{b}_{k_2}\}$ such that $R_\Phi(\pi(\mathfrak{v}_i)) = \varnothing$.

For every moment from A_1 apply Lemma 5.3.17. Then for each $\mathfrak{v}_i$ we obtain a forest $\mathfrak{F}_i$ with a Φ-morphism $\rho_i^1 : |\mathfrak{m}_i| \to \mathfrak{F}_i$ and a naively monotone relation $R_i^1 \subseteq |\mathfrak{v}_i| \times |\mathfrak{F}_i|$.

Now consider the moments from A_2. For each moment $\mathfrak{b}_i$, we can take a copy of it and define empty relation R_i^2, and take Φ-morphism $\rho_i^2 = \pi \restriction_{|\mathfrak{b}_i|}$. Then R_i^2 relation is trivially naively monotone.

Since every forest $\mathfrak{F}_i$ is just a set of moments, we can take the disjoint union of these forests along with copies of $\mathfrak{b}_i$ and obtain a forest $\mathfrak{F}$ with a Φ-morphism $\rho : |\mathfrak{F}_I| \to |\mathfrak{F}|$, where

$$\rho = \bigsqcup_{i=1}^{k_1} \rho_i^1 \sqcup \bigsqcup_{i=1}^{k_2} \rho_i^2$$

and $R_I \subseteq |\mathfrak{F}_I| \times |\mathfrak{F}|$ such that:

$$R_I = \bigsqcup_{i=1}^{k_1} R_i^1 \sqcup \bigsqcup_{i=1}^{k_2} R_i^2.$$

Thus, we obtain the desired story by adding $\mathfrak{F}$ to $\mathbf{S}'$. $\square$

This proposition shows that $\mathcal{DRL}$ is complete.

Theorem 5.3.19. $\mathcal{DRL}$ is complete with respect to the class of all finite naively monotone dynamic Kripke frames.

Proof. Assume that $\mathcal{DRL} \nvdash \varphi$. Then by Lindenbaum's lemma there is a point w in W_c, s.t. $\mathfrak{M}, w \nvDash \varphi$. Take I to be the $\langle \mathrm{R} \rangle$-depth of φ and $\Phi = Sub(\varphi)$. By lemma 5.3.18, we have that there is a story $\mathbf{S}$ of duration I, with a Φ-morphism $\pi : \mathbf{S} \to W_c$ s.t. $w = \pi(v)$, where v is the root of the moment in the story $\mathbf{S}$. Therefore, by Lemma 5.3.13, we have that $\mathbf{S}, v \nvDash \varphi$. □

The proof of completeness for logic $\mathcal{DPL}$ requires a slightly different argument. This is because the proof of Lemma 5.3.17 does not ensure functionality and totality of R in case $F \in \mathsf{L}$. That is why we prove a different version of Lemma 5.3.17 for logics that extend $\mathcal{DPL}$. Our proof is similar to the proof of Lemma 6.15 [FM21].

Definition 5.3.20. Let $\mathcal{F}_1 = (W_1, \sqsubseteq_1)$, $\mathcal{F}_2 = (W_2, \sqsubseteq_2)$ be two trees with roots r_1 and r_2 respectively. Then $f : W_1 \to W_2$ is called root-preserving if $f(r_1) = r_2$.

Lemma 5.3.21. Let $\mathsf{L} \supseteq \mathcal{DPL}$ and $\mathfrak{M}_{\mathsf{L}} = (W_{\mathrm{c}}, \sqsubseteq_{\mathrm{c}}, f_{\mathrm{c}}, \nu_{\mathrm{c}})$ be its canonical model. Suppose that there exists a Φ-morphism $\pi : |\mathfrak{m}| \to W_c$. Then there exist a moment $\mathfrak{n}$, a Φ-morphism $\rho : |\mathfrak{n}| \to W_c$ and a monotone map $g : |\mathfrak{m}| \to |\mathfrak{n}|$, such that $f_c \circ \pi = \rho \circ g$ and $g(r_{\mathfrak{m}}) = r_{\mathfrak{n}}$.

Proof. We proceed by the induction on height of $\mathfrak{m}$.

Assume that for every moment $\mathfrak{n}$, s.t. $\mathbf{height}(\mathfrak{n}) < \mathbf{height}(\mathfrak{m})$ the statement holds. Take the root r of the moment $\mathfrak{m}$.

Take $\mathcal{A} = \{a_1, \dots, a_m\}$ the set of all strict successors of r. For each a_i we can take the generated by a_i submodel α_i. Then α_i is a moment, s.t. $\mathbf{height}(\alpha_i) < \mathbf{height}(\mathfrak{m})$. Thus, there is a sequence of moments $(\alpha_i')_{i \leq m}$, monotone root-preserving maps $f_i : |\alpha_i| \to |\alpha_i'|$, and Φ-morphisms $\rho_n : |\alpha_i'| \to W_c$ such that $f_c \circ \pi_i = \rho_i \circ g_i$, where $\pi_i = \pi \upharpoonright_{|\alpha_i|}$.

For every $v \sqsupseteq_\Phi f_c(\pi(r))$, by Lemma 5.3.16 there are $\mathfrak{b}_v$ and a Φ-morphism $\rho_v : |\mathfrak{b}_v| \to W_c$ mapping the root of $\mathfrak{b}_v$ to v. Take $\vec{\mathfrak{a}} = (\alpha_i')_{i \leq m}$ and $\vec{\mathfrak{b}} = (\mathfrak{b}_v)_v$. Then define $\mathfrak{n} = \binom{\vec{\mathfrak{a}} * \vec{\mathfrak{b}}}{f_c(\pi(r))}$. Define $\rho : |\mathfrak{n}| \to W_c$ and $g : |\mathfrak{n}| \to |\mathfrak{m}|$ as follows:

$$\rho(w) = \begin{cases} \rho_v(w), & \text{if } w \in \mathfrak{b}_v; \\ \rho_i(w), & \text{if } w \in |\alpha_i'| \\ f_c(\pi(r)), & \text{if } w = f_c(\pi(r)). \end{cases}$$

$$g(w) = \begin{cases} f_c(w), & \text{if } w =; \\ \rho_i(w), & \text{if } w \in |\alpha_i'|; \end{cases}$$

It is easy to see that ρ is a Φ-morphism and g is a monotone map with $f_c \circ \pi = \rho \circ g$.

□

Proposition 5.3.22. Let $\mathsf{L} \supseteq \mathcal{DPL}$ be a logic and $\mathfrak{M}_{\mathsf{L}} = (W_{\mathrm{c}}, \sqsubseteq_{\mathrm{c}}, g_{\mathrm{c}}, \nu_{\mathrm{c}})$ be its canonical model. Assume that $I < \omega$ and $w \in W_{\mathrm{c}}$. Then there is a story $\mathbf{S}$ of duration I such that:

1. For every $i \leq I$, the forest $\mathfrak{F}_i$ in story $\mathbf{S}$, consists of one moment;
2. For every $i < I$, the naively monotone relation $R_i \subseteq |\mathfrak{F}_i| \times |\mathfrak{F}_{i+1}|$ is a monotone root preserving function;

and a dynamic Φ-morphism $\pi : |\mathbf{S}| \to W_c$ with $w = \pi(C_0)$

Proof. The proof is by induction on I. For $I = 0$, we use Lemma 5.3.16. For the inductive step, assume that a story $\hat{\mathfrak{S}}$ is of depth I and a dynamic Φ-morphism π exists. Take the forest $\mathfrak{F}_I$. It is a moment $\mathfrak{m}$. Thus, by Lemma 5.3.21 , there is a moment $\mathfrak{n}$, monotone root-preserving map $f_I : |\mathfrak{m}| \to |\mathfrak{n}||$, and Φ-morphism $\pi' : |\mathfrak{n}| \to W_{\mathrm{c}}$ such that $f_c \circ \pi = \pi' \circ g$. We define $\mathbf{S}$ by adding $\mathfrak{n}$ to $\mathbf{S}$ so we obtain the desired story. □

It follows that every formula satisfiable in a point in the canonical model of $\mathsf{L} \supseteq \mathcal{DPL}$ is also satisfied on a finite story.

Theorem 5.3.23. $\mathcal{DPL}$ is complete with respect to the class of all finite monotone dynamic Kripke frames.

Proof. The proof is analogous to the proof of Theorem 5.3.19, but using Lemma 5.3.22 to obtain the right story $\mathbf{S}$ with a Φ-morphism. □

5.4 Constructing a polyhedron

We have proven that $\mathcal{DRL}$ and $\mathcal{DPL}$ are complete with respect to the corresponding classes of Kripke frames. We devote this section we devote to the method of constructing a polyhedron from a finite dynamic Kripke frame. We will use the technique outlined in [Ada+22]. This technique employs the tool from combinatorial and polyhedral geometry, the *nerve* of a poset.

Definition 5.4.1. Suppose $\mathcal{F} = (W, \sqsubseteq)$ is a Kripke frame. Then the *nerve* $\mathcal{N}(\mathcal{F})$ of $\mathcal{F}$ is a tuple $(\mathcal{C}, \subseteq)$, where $\mathcal{C} = \{A \mid A \text{ is a chain in } \mathcal{F}\}$.

Since $\subseteq$ is a poset relation, $\mathcal{N}(\mathcal{F})$ is a Kripke frame. We will use nerves to construct a polyhedron, and that is why we want this relation to be dynamic in the sense of definition 3.3.1. That is why we use the following definition.

Definition 5.4.2. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic Kripke frame and $C \subseteq \mathcal{F}$ be a chain. Define $P \subseteq \mathcal{C} \times \mathcal{C}$ in the following way:

$$
\begin{aligned}
CPC' \text{ iff } C' \subseteq R(C), \text{ and} \\
\text{there is a monotonic surjective function } h : C \to C' \text{ with } h(y) \in R(y).
\end{aligned}
$$

Lemma 5.4.3. Let $\mathcal{F} = (w, \sqsubseteq, f)$ be a dynamic monotone Kripke frame. Then $P \subseteq \mathcal{C} \times \mathcal{C}$ is a function.

Proof. We have to show that P is functional and total. Take some $C \in \mathcal{C}$. Then $f(C)$ is a chain since f is monotone. By definition $f \restriction_C : C \to f(C)$ is surjective and monotone. Assume that there is $C'' \subseteq f(C)$ such that CPC''. Then there is a monotone surjective function $h : C \to C''$ such that $h(c) = f(c)$ for every $c \in C$. Thus $C'' = f(C)$. □

Thus, the definition of a nerve of a dynamic Kripke frame is:

Definition 5.4.4. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic Kripke frame. Then we call $\mathcal{N}(\mathcal{F}^{\mathcal{D}}) = (\mathcal{C}, \subseteq, P)$ the *nerve of* $\mathcal{F}^d$ if

1. $(\mathcal{C}, \subseteq)$ is the nerve of $(W, \sqsubseteq)$;

2. P is the relation from Definition 5.4.2.

We can show that P is naïvely monotone:

Lemma 5.4.5. Let $\mathcal{F}^{\mathcal{D}} = (\mathcal{W}, \sqsubseteq, \mathcal{R})$ be a dynamic Kripke frame. Then the relation P on its nerve $\mathcal{N}(\mathcal{F}^{\mathcal{D}}) = (\mathcal{C}, \subseteq, P)$ is naïvely monotone with respect to $\subseteq$.

Proof. Assume that $C_1 \subseteq C_2$ and $C_2 P C_2'$. Then $C_2' \subseteq R(C_2)$ is a chain, with a monotone surjective function $h : C_2 \to C_2'$. Take $h' = h \restriction_{C_1}$. Then h' is a surjective monotone function. Since $h'(C_1) \subseteq C_2'$, we have that it is a chain. As $h' \subseteq h$, then for every $y \in C_1$ we have that $h'(y) \in R(y)$. Thus, $C_1 P h'(C_1)$. □

Before moving further, we define a special p-morphism for nerves and Kripke frames employed in [Ada+22].

Definition 5.4.6. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic monotone Kripke frame and $\mathcal{N}(\mathcal{F}^{\mathcal{D}}) = (\mathcal{C}, \subseteq, P)$ be its nerve. Then **max** : $\mathcal{C} \to W$ is a function that sends a chain to its maximal element.

Lemma 5.4.7. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic monotone Kripke frame and $\mathcal{N}(\mathcal{F}^{\mathcal{D}}) = (\mathcal{C}, \subseteq, P)$ be its nerve. Then **max** : $\mathcal{N}(\mathcal{F}) \to \mathcal{F}$ is a p-morphism.

Proof. See [Ada+22, Proposition 2.3]. □

We will show that **max** is a dynamic p-morphism for every dynamic naïvely monotone frame.

Lemma 5.4.8. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic monotone Kripke frame. Then CPC' implies $\mathbf{max}(C) R\, \mathbf{max}(C')$.

Proof. If CPC', take the surjective monotone map $h : C \to C'$ with $h(c) \in R(C)$ for every $c \in C$. Then $\mathbf{max}(C') \in R(C)$ and at the same time $h(\mathbf{max}(C)) \in R(\mathbf{max}(C))$. Since h is monotone, we have that $h(\mathbf{max}(C)) = \mathbf{max}(C')$. □

Lemma 5.4.9. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic naïvely monotone Kripke frame. Then $\mathbf{max}(C) Ry$ implies that there is $\mathbf{max}(C') = y$ and CPC'.

Proof. We prove the statement by induction on the length $\updownarrow$ of the chain C.

$l = 1$.

If $|C| = 1$, then define $C' = \{y\}$. Then clearly CPC'.

$l \curvearrowright l + 1$.

Assume that the chain C consists of elements $\{z_0, \ldots, z_l\}$, ordered by their indexes, and $z_l Ry$. Take the first $l - 1$ elements $S = \{z_0, \ldots, z_{l-1}\}$. Since $z_{l-1} \sqsubseteq z_l$ and $z_l Ry$, then by naïve monotonicity of R we have that there is an element y' such that $z_{l-1} Ry'$ and $y' \sqsubseteq y$. Hence, there exists a chain S' such that SPS' and $\mathbf{max}(S') = y'$. Take the monotone surjective function $h : S \to S'$. Since $y' \sqsubseteq y$, we have that $C^{'} = S' \cup \{y\}$ is a chain and $h' = h \cup (z_l, y)$ is a subjective monotone function such that $h(c) \in R(C)$ for every $c \in C$. We conclude that CPC'. □

We conclude this part with the following lemma.

Lemma 5.4.10. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic monotone or naïvely monotone frame then $\mathcal{N}(\mathcal{F}^{\mathcal{D}}) \vDash \varphi$ implies $\mathcal{F}^d \vDash \varphi$.

Proof. By the last two lemmas, we conclude that $\mathbf{max} : \mathcal{N}(\mathcal{F}^{\mathcal{D}}) \to \mathcal{F}^d$ is a dynamic p-morphism, and therefore by Lemma 5.3.9 the result follows. □

5.4.1 From nerves to polyhedra

Now we will elaborate on how to construct a polyhedron from the nerve $\mathcal{N}(\mathcal{F})$. This construction was proposed in [Ada+22] regarding polyhedra without dynamic function. For every Euclidean space $\mathbb{R}^n$ let $e_1, \ldots, e_n$ be its standard basis.

Definition 5.4.11. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite Kripke frame, enumerate it as $W = \{x_1, \ldots, x_n\}$. Take the Euclidean space $\mathbb{R}^n$ and a bijection $o : W \to \{e_1, \ldots, e_n\} : x_i \mapsto e_i$.

The *simplicial complex induced by* $\mathcal{F}^d$ is the set of simplexes:

$$\mathcal{K} := \{Conv\{o(C)\} \mid C \in \mathcal{N}(\mathcal{F})\},$$

and the dynamic relation $P^* \subseteq \mathcal{K} \times \mathcal{K}$ is defined as follows:

$$Conv(o(C_1))P^*Conv(o(C_2)) \text{ iff } C_1 P C_2.$$

Since $\mathcal{K}$ is a simplicial complex, we have that $|\mathcal{K}|$ is a polyhedron and $\widetilde{\mathcal{K}}$ is as before the partition of $|\mathcal{K}|$.

Lemma 5.4.12. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic Kripke frame and $o : W \to \{e_1, \ldots, e_n\}$ a bijection. Then $o^* : \mathcal{C} \to \mathcal{K} : C \mapsto Conv(o(C))$ is an isomorphism.

Proof. Clearly, o^* is surjective. Note that if $C_1 \neq C_2$ are different then $Conv(o(C_1)) \neq Conv(o(C_2))$, hence o^* is injective.

Assume that $C_1 \subseteq C_2$. Then $Conv(o(C_1)) \subseteq Conv(o(C_2))$. Therefore $Conv(o(C_1)) \preccurlyeq Conv(o(C_2))$.

If $Conv(o(C_1)) \preccurlyeq Conv(o(C_2))$, then $Conv(o(C_1)) \subseteq Conv(o(C_2))$. Hence, $o(C_1) \subseteq o(C_2)$, since $o(C_2)$ are from the basis. Thus, $C_1 \subseteq C_2$. $\square$

Definition 5.4.13. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic Kripke frame. Then define $\widetilde{P}^* \subseteq \widetilde{\mathcal{K}} \times \widetilde{\mathcal{K}}$ in the following way:

$$\widetilde{\sigma}\widetilde{P}^*\widetilde{\tau} \text{ iff } \sigma P^* \tau.$$

Definition 5.4.14. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic Kripke frame. Then we call $\nabla\mathcal{F}^d = (\widetilde{K}, \precsim, \widetilde{P}^*)$ a simplicial structure induced by $\mathcal{F}^d$ if:

1. $\widetilde{\mathcal{K}}$ is the set of relative interiors of simplexes in $\mathcal{K}$;
2. $\widetilde{P}^*$ is the relation from Definition 5.4.13.

Thus, we obtain that the nerve $\mathcal{N}(\mathcal{F}^{\mathcal{D}})$ is isomorphic to simplicial structure $\nabla\mathcal{F}^d$.

Proposition 5.4.15. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic Kripke frame. Then for its nerve $\mathcal{N}(\mathcal{F}^{\mathcal{D}}) = (\mathcal{C}, \subseteq, P)$ and $\nabla\mathcal{F}^d = (\widetilde{K}, \precsim, \widetilde{P}^*)$ its simplicial structure we have:

$$\mathcal{N}(\mathcal{F}^{\mathcal{D}}) \vDash \varphi \text{ iff } \nabla\mathcal{F}^d \vDash \varphi,$$

for every formula φ in $\mathcal{L}_{\square,\langle \mathrm{R}\rangle}$.

Proof. Since $\mathcal{N}(\mathcal{F}^{\mathcal{D}})$ is isomorphic to $\nabla\mathcal{F}^d$, then we have that there are two surjective dynamic p-morphisms: $o^* : \mathcal{C} \to \widetilde{\mathcal{K}}$ and $(o^*)^{-1} : \widetilde{\mathcal{K}} \to \mathcal{C}$. $\square$

To obtain a dynamic system from $(\widetilde{\mathcal{K}}, \precsim, \widetilde{P}^*)$ we have to define the relation R^* on polyhedron $|\mathcal{K}|$. Here we will again have to work with two different cases - namely, the case of dynamic naïvely monotone frames and the case of dynamic monotone frames. The latter R^* must be a polyhedral function. We divide the further discussion into two subsections, the first of which is devoted to constructing a dynamic relational system from a dynamic naïvely monotone frame, and the second to constructing a dynamic polyhedral system from a dynamic monotone frame.

5.4.2 From Dynamic naïvely monotone frames to Dynamic relational systems

When we start with a dynamic naïvely monotone frame $\mathcal{F}^d = (W, \sqsubseteq, R)$, we have that its simplicial complex $(\widetilde{\mathcal{K}}, \precsim, \widetilde{P}^*)$ is also a dynamic naïvely monotone frame, as proved in Lemma 5.4.12. So, let us define the relation on $|\mathcal{K}|$ in the following way.

Definition 5.4.16. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic Kripke frame and $\mathcal{K}$ be a simplicial complex induced by $\mathcal{F}^d$. Define $R^* \subseteq |\mathcal{K}| \times |\mathcal{K}|$ in the following way:

$$xR^*y \text{ iff } \widetilde{\sigma}^x \widetilde{P}^* \widetilde{\sigma}^y.$$

Observe that the following holds with definition 5.4.16.

Lemma 5.4.17. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic Kripke frame. Then for relation R^* from Definition 5.4.16 the following is true:

$$\widetilde{\sigma}\widetilde{P}^*\widetilde{\tau} \text{ iff } \exists x \in \widetilde{\sigma}, y \in \widetilde{\tau} : xR^*y,$$

Proof. Assume that $\widetilde{\sigma}\widetilde{P}^*\widetilde{\tau}$. Then since the relative interior of every simplex is nonempty, we have that there is $x \in \widetilde{\sigma}$ and there is $y \in \widetilde{\tau}$. By fact that $\widetilde{\mathcal{K}}$ is the partition, we have that $\widetilde{\sigma}^x = \widetilde{\sigma}$ and $\widetilde{\sigma}^y = \widetilde{\tau}$. Thus, xR^*y.

The other direction follows directly from the R^* definition. $\square$

To prove that $(|\mathcal{K}|, \mathcal{K}, R^*)$ is a dynamic relational system, we need to show that:

$$(R^*)^{-1}(\widetilde{\sigma}) \in \mathfrak{U}(\mathcal{K}); \tag{$*$}$$

$$(R^*)^{-1}(P') \in Sub_{\mathcal{K}}(P), \text{ whenever } P' \in Sub_{\mathcal{K}}(P). \tag{$**$}$$

First, we will show that $(*)$ holds, and then by Lemma 3.3.13, we will obtain $(**)$. To prove the first item it is sufficient to show that $(R^*)^{-1}(\widetilde{\sigma}) = \bigcup((\widetilde{P}^*)(\widetilde{\sigma}))$, since $\widetilde{P}^*(\widetilde{\sigma}) \subseteq \widetilde{\mathcal{K}}$.

Lemma 5.4.18. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic Kripke frame and $\nabla\mathcal{F}^d = (\widetilde{K}, \precsim, \widetilde{P}^*)$ its simplicial structure. Then we have:

$$(R^*)^{-1}(\widetilde{\sigma}) = \bigcup(\widetilde{P}^*)^{-1}(\widetilde{\sigma}).$$

Proof. $(\subseteq)$ Take $x \in (R^*)^{-1}(\widetilde{\sigma})$. Then there is $y \in \widetilde{\sigma}$ such that xR^*y. But then by definition of R^* and the fact that $\widetilde{\sigma} = \widetilde{\sigma}^y$ we have that $\widetilde{\sigma}^x \widetilde{P}^* \widetilde{\sigma}$. Hence, we obtain $\widetilde{\sigma}^x \in (\widetilde{P}^*)^{-1}(\widetilde{\sigma})$, and therefore $x \in \bigcup((\widetilde{P}^*)^{-1}(\widetilde{\sigma}))$.

$(\supseteq)$ For the other direction take $x \in \bigcup(\widetilde{P}^*)^{-1}(\widetilde{\sigma})$. Then for $\widetilde{\sigma}^x$ it is true that $\widetilde{\sigma}^x \widetilde{P}^* \widetilde{\sigma}$. Hence, by the definition of R^* we have that xR^*y for some $y \in \widetilde{\sigma}$. We conclude that $x \in (R^*)^{-1}(\widetilde{\sigma})$. $\square$

Definition 5.4.19. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic Kripke frame. We call $\mathcal{D} = (|\mathcal{K}|, \mathcal{K}, R^*)$ the dynamic realisation of $\mathcal{F}^d = (W, \sqsubseteq, R)$ if:

1. $\mathcal{K}$ is the simplicial complex induced by $\mathcal{F}^d$;
2. R^* is the relation on $|\mathcal{K}|$ from Definition 5.4.16.

Proposition 5.4.20. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a finite dynamic Kripke frame. Then its dynamic realization $\mathcal{D} = (|\mathcal{K}|, \mathcal{K}, R^*)$ is a dynamic relational system, with the encoding $\mathcal{F}(\mathcal{D}) = (\widetilde{\mathcal{K}}, \precsim, \widetilde{P}^*)$.

Proof. By lemma 5.4.18 we obtain that for every $\widetilde{\sigma} \in \widetilde{\mathcal{K}}$: $(R^*)^{-1}(\widetilde{\sigma}) \in \mathfrak{U}(\mathcal{K})$. Then by Lemma 5.4.17 and Lemma 3.3.13 we have that $(R^*)^{-1}(P') \in Sub_{\mathcal{K}}(P)$, whenever $P' \in Sub_{\mathcal{K}}(P)$. $\square$

Theorem 5.4.21. For a finite dynamic naïvely monotone frame $\mathcal{F}^d = (W, \sqsubseteq, R)$ and its dynamic realisation $\mathcal{D} = (|\mathcal{K}|, \mathcal{K}, R^*)$:

$$\mathcal{D} \vDash \varphi \Rightarrow \mathcal{F}^d \vDash \varphi,$$

for every φ in $\mathcal{L}_{\Box\langle\mathrm{R}\rangle}$.

Proof. By lemma 5.1.12 we have that $\mathcal{D} \vDash \varphi$ iff $\mathcal{F}(\mathcal{D}) \vDash \varphi$. Since $\mathcal{N}(\mathcal{F}^{\mathcal{D}})$ is isomorphic to $\mathcal{F}(\mathcal{D}) = (\widetilde{\mathcal{K}}, \precsim, \widetilde{P}^*)$, we obtain that $\mathcal{F}(\mathcal{D}) \vDash \varphi$ iff $\mathcal{N}(\mathcal{F}^{\mathcal{D}}) \vDash \varphi$. Using lemma 5.4.10 we have that $\mathcal{N}(\mathcal{F}^{\mathcal{D}}) \vDash \varphi \Rightarrow \mathcal{F}^d \vDash \varphi$ for every φ in $\mathcal{L}_{\Box\langle\mathrm{R}\rangle}$. Hence, $\mathcal{D} \vDash \varphi \Rightarrow \mathcal{F}^d \vDash \varphi$. $\square$

5.4.3 From Dynamic monotone frames to Dynamic polyhedral systems

Fix for this section some dynamic monotone frame $\mathcal{F}^d = (W, \sqsubseteq, R)$. As mentioned in Lemma 5.4.3, $P \subseteq \mathcal{N}(\mathcal{F}^{\mathcal{D}}) \times \mathcal{N}(\mathcal{F}^{\mathcal{D}})$ becomes a function. Then by Lemma 5.4.5, we have that P is a monotone function. Observe, that P is determined by its values on the chains of length 1 $\big($i.e. atoms of $\mathcal{N}(\mathcal{F}^{\mathcal{D}})\big)$. Then we can define a function $f^{'} : \mathcal{K} \to \mathcal{K}$:

Definition 5.4.22. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic monotone frame and $\mathcal{K}$ be a simplicial complex induced by $\mathcal{F}^d$. Define $f^{'} : \mathcal{K} \to \mathcal{K}$ as follows:

$$f^{'}(o^*(C)) = o^*(P(C)), \tag{5.2}$$

The $f^{'}$ function is naturally translated to the function $f : \widetilde{\mathcal{K}} \to \widetilde{\mathcal{K}}$:

$$f(\widetilde{\sigma}) = \widetilde{\tau} \text{ iff } f'(\sigma) = \tau \tag{$\star$}$$

We will show that f is monotone.

Lemma 5.4.23. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic monotone frame. The function $f : \widetilde{\mathcal{K}} \to \widetilde{\mathcal{K}}$ defined as $\star$ is monotone.

Proof. We will prove it through the proof that f' is monotone. Assume that $\sigma \preccurlyeq \tau$, then for the corresponding chains C_σ, C_τ such that $o^*(C_\sigma) = \sigma$ and $o^*(C_\tau) = \tau$, we have that $C_\sigma \subseteq C_\tau$, and therefore $P(C_\sigma) \subseteq P(C_\tau)$, because P is monotone. Since $Conv$ is monotone, we have that $Conv(o(P(C_\sigma))) \subseteq Conv(o(P(C_\tau)))$. Since both $Conv(o(P(C_\sigma)))$ and $Conv(o(P(C_\tau)))$ are simplexes, we have that $o^*(P(C_\sigma)) \preccurlyeq o^*(P(C_\tau))$. Thus, $f'(\sigma) \preccurlyeq f'(\tau)$. □

We observe that f' is completely determined by its values on the basis.

Lemma 5.4.24. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic monotone frame. Then for $f' : \mathcal{K} \to \mathcal{K}$ from Definition 5.4.22 we have that:

$$f'(Conv(e_1, \ldots, e_n)) = Conv(f'(e_1), \ldots, f'(e_n)).$$

Proof. Let us take $Conv(e_{i_1}, \ldots, e_{i_n}) \in \mathcal{K}$. Then $\{e_{i_1}, \ldots, e_{i_n}\} = o(C)$ for some $C = \{x_{i_1} \ldots, x_{i_n}\}$, where $C \in \mathcal{C}$. By the definition of f' we have that $f'(o^*(C)) = o^*(P(C))$. Then:

$$\begin{aligned} o^*(P(C)) &= Conv(o(P(\{x_{i_1} \ldots, x_{i_n}\}))) \text{ (definition of } o^*) \\ &= Conv(\{o(P(x_{i_1})), \ldots, o(P(x_{i_n}))\}) \text{ (since } P \text{ is a function)} \\ &= Conv(\{f'(e_{i_1}), \ldots, f'(e_{i_n})\}) \text{ (since } Conv(o(P(x_{i_j}))) = o(P(x_{i_j}))) \end{aligned}$$

□

So, we have that $(\widetilde{\mathcal{K}}, \precsim, \tilde{f})$ is a dynamic monotone frame. Now from it, we will construct a dynamic polyhedral system using $|\mathcal{K}|$ as the domain.

Definition 5.4.25. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic monotone frame, $\mathcal{K}$ be its simplicial complex and $f' : \mathcal{K} \to \mathcal{K}$ be a function defined in 5.4.22. We call $f^* : |\mathcal{K}| \to |\mathcal{K}|$ the *polyhedral protagonist for* $\mathcal{F}^d$ if:

$$f^*(x) = \sum_{i=0}^{n} \lambda_i f'(e_i), \text{ where } x = \sum_{i=0}^{n} \lambda_i e_i. \tag{5.3}$$

Now we have to show two facts:

1. f^* is continuous;
2. $(f^*)^{-1}(P') \in Sub_{\mathcal{K}}(|\mathcal{K}|)$, for every $P' \in Sub_{\mathcal{K}}(|\mathcal{K}|)$.

In other words, we have to show that f^* is polyhedral in order to obtain that $(|\mathcal{K}|, \mathcal{K}, f^*)$ is a dynamic polyhedral system. The second item will follow from lemma 3.3.13, if we show that $(f^*)^{-1}(\widetilde{\sigma}) \in \mathfrak{U}(\mathcal{K})$. For this, we will need several technical lemmas.

Lemma 5.4.26. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic monotone frame, $\mathcal{K}$ be its simplicial complex, f be a function defined as in $(\star)$ and $f^* : |\mathcal{K}| \to |\mathcal{K}|$ be a polyhedral protagonist for $\mathcal{F}^d$. Then for all $\widetilde{\sigma}, \widetilde{\tau} \in \widetilde{\mathcal{K}}$: $f^*(\widetilde{\sigma}) \subseteq \widetilde{\tau}$ iff $f(\widetilde{\sigma}) = \widetilde{\tau}$.

Proof. $(\Rightarrow)$ Assume that $x \in \widetilde{\sigma}$. Then by definition of σ we have that $\sigma = Conv(e_{i_1}, \dots, e_{i_n})$ of elements from the basis. Then $x = \lambda_{i_1} e_{i_1} + \dots, \lambda_{i_n} e_{i_n}$ is the convex combination with no lambda being equal to 0. By the definition of f^*:

$$f^*(x) = \lambda_{i_1} f'(e_{i_1}) + \dots, \lambda_{i_n} f'(e_{i_n}).$$

By assumption we have that $f^*(x) \in \widetilde{\tau}$. Since τ is the smallest simplex such that $f^*(x) \in \widetilde{\tau}$ we have that τ is the convex hull on points $f'(e_{i_1}), \dots, f'(e_{i_n})$. By Lemma 5.4.24: $f'(\sigma) = \tau$. Hence, $f(\widetilde{\sigma}) = \widetilde{\tau}$.

$(\Leftarrow)$ If $f(\widetilde{\sigma}) = \widetilde{\tau}$, then $f'(\sigma) = \tau$. By the definition of σ in $\mathcal{K}$ we have that $\sigma = o^*(\{x_{i_1}, \dots, x_{i_n}\})$ for some $\{x_{i_1}, \dots, x_{i_n}\}$ chain in W. Then $f'(\sigma) = Conv(f'(e_{i_1}), \dots, f'(e_{i_n})) = \tau$ by Lemma 5.4.24. Taking $x \in \widetilde{\sigma}$ we have that $x = \lambda_{i_1} e_{i_1} + \dots, \lambda_{i_n} e_{i_n}$, for every $\lambda_{i_j} > 0$. Thus, $f^*(x) = \lambda_{i_1} f'(e_{i_1}) + \dots, \lambda_{i_n} f'(e_{i_n})$. We conclude that $f^*(x) \in \widetilde{\tau}$. $\square$

Lemma 5.4.27. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic monotone frame, $\mathcal{K}$ be its simplicial complex, f be a function defined as in $(\star)$ and $f^* : |\mathcal{K}| \to |\mathcal{K}|$ be a polyhedral protagonist for $\mathcal{F}^d$. Then for all $x \in |\mathcal{K}|$, if $f^*(x) \in \widetilde{\sigma}$, then $f^*(\widetilde{\sigma}^x) \subseteq \widetilde{\sigma}$.

Proof. Assume $f^*(x) \in \widetilde{\sigma}$ for $x = \lambda_{i_1} e_{i_1} + \ldots, \lambda_{i_n} e_{i_n}$, where all $\lambda_{i_j} > 0$. Since σ is the smallest simplex such that $f^*(x) \in \widetilde{\sigma}$, we have that $\widetilde{\sigma} = Conv(f^*(e_{i_1}), \ldots f^*(e_{i_n}))$, since $f'(e_j) = f^*(e_j)$ for every e_j. Then for every $y \in \widetilde{\sigma}^x$ we have that $y = \lambda'_{i_1} e_{i_1} + \ldots, \lambda'_{i_n} e_{i_n}$, where every $\lambda_{i_j} > 0$. Thus, $f^*(y) = \lambda'_{i_1} f^*(e_{i_1}) + \ldots, \lambda'_{i_n} f^*(e_{i_n})$, and since $f(e_{i_j}) = f^*(e_{i_j})$, we conclude that $f^*(y) \in \widetilde{\sigma}$. □

From the lemmas above, it follows that:

Proposition 5.4.28. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic monotone frame, $f : \widetilde{\mathcal{K}} \to \widetilde{\mathcal{K}}$ a function $\star$ and f^* dynamic protagonist of $\mathcal{F}^d$. Then:

$$(f^*)^{-1}(\widetilde{\sigma}) = \bigcup f^{-1}(\widetilde{\sigma}).$$

Proof.

$$\begin{aligned}
(f^*)^{-1}(\widetilde{\sigma}) &= \bigcup \{\widetilde{\tau} \mid f^*(\widetilde{\tau}) \subseteq \widetilde{\sigma}\} \text{ (by Lemma 5.4.27)} \\
&= \bigcup \{\widetilde{\tau} \mid f(\widetilde{\tau}) = \widetilde{\sigma}\} \text{ (by lemma } 5.4.26) \\
&= \bigcup f^{-1}(\widetilde{\sigma}).
\end{aligned}$$

□

Lemma 5.4.29. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic monotone frame, $f : \widetilde{\mathcal{K}} \to \widetilde{\mathcal{K}}$ a function $\star$ and f^* dynamic protagonist of $\mathcal{F}^d$. Then $(f^*)^{-1}(P') \in Sub_{\mathcal{K}}(|\mathcal{K}|)$, for every $P' \in Sub_{\mathcal{K}}(|\mathcal{K}|)$.

Proof. Combining Lemma 3.3.13 and Lemma 5.4.28. □

Let us now prove that f^* is continuous.

Lemma 5.4.30. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic monotone frame. Then for the dynamic protagonist f^* is a continuous map on $|\mathcal{K}|$.

Proof. We will need some preparation to show that f^* is continuous.

The standard basis $E = \{e_1, \ldots, e_n\}$ is fixed, so take $f^*(E) = \{e'_1, \ldots, e'_r\}$.

Then, if $x = \sum\limits_{i=1}^{n} \lambda_i e_i$, we have that $f^*(x) = \sum\limits_{i=1}^{n} \lambda_i f^*(e_i)$ and $\sum\limits_{i=1}^{n} \lambda_i f^*(e_i) = \sum\limits_{i=1}^{r} (\lambda_{i_1} + \ldots + \lambda_{i_p}) e'_i$. Take $\{\lambda_{i_1}, \ldots, \lambda_{i_p}\}$ for all lambdas, which are coefficients for e'_i. Define $\mu_i = \sum\limits_{j=0}^{p} \lambda_{i_j}$. Hence, we can rewrite $f^*(x)$ in the following way:

$$f^*(x) = \sum_{k=1}^{r} \mu_k e_k^{'}.$$

Observe that $\sum_{i=1}^{n} \lambda_i = 1$ and every $\lambda_i \in [0,1]$ iff $\sum_{k=1}^{p} \mu_k = 1$ and $\mu_k \in [0,1]$, so in the future proof we will not take into account this condition for points.

First, let us recall that ρ is the Euclidean metric in our space. Suppose now that $B_\varepsilon(o) \subseteq |\mathcal{K}|$ is an open ball of radius ε with center o. Denote it as $V = B_\varepsilon(o)$. We will show that for every $x \in (f^*)^{-1}(V)$, x is contained in an open set $U \subseteq (f^*)^{-1}(V)$. By the definition of convex hull, $x = \sum_{i=1}^{n} \lambda_i e_i$. Therefore, $f^*(x) = \sum_{i=1}^{r} \mu_i e_i^{'}$. Since $f^*(x) \in B_\varepsilon(o)$, we have that $\rho(o, f^*(x)) = t < \varepsilon$. We will define an open ball $B_\delta(x)$, in such a way that $f^*(B_\delta(x)) \subseteq V$. So we need the following property:

$$\rho(x,z) < \delta \Rightarrow \rho(o, f^*(z)) < \varepsilon, \text{ for every } z \in V.$$

This inequality will be preserved if $\rho(f^*(x), f^*(z)) < \varepsilon - t$. Indeed, in this case: $\rho(o, f^*(z)) \leq \rho(o, f^*(x)) + \rho(f^*(x), f^*(z)) < t + \varepsilon - t = \varepsilon$. Let us denote $\varepsilon - t$ with $\varepsilon^{'}$.

We will define δ in such a way that

$$\rho(f^*(x), f^*(z)) < \varepsilon'.$$

First, by the definition of f^*, $f^*(z)$ is:

$$f^*(z) = \sum_{i=1}^{r} \mu_i^{'} e_i^{'},$$

where $\mu_i^{'} = \sum_{j=1}^{p} \lambda_{i_j}^{'}$, and $\lambda_{i_j}^{'}$ are coefficients of z.

Hence, we have that $\rho(f^*(x), f^*(z)) < \varepsilon^{'}$, whenever $\mu_i^{'} \in (\mu_i - \varepsilon^{'}/\sqrt{r}, \mu_i + \varepsilon^{'}/\sqrt{r})$. Rewriting $\mu_i^{'}$ with the sums of lambdas, we obtain the following fact $\sum_{j=0}^{p} \lambda_{i_j}^{'} \in ((\sum_{j=0}^{p} \lambda_{i_j}) - \varepsilon^{'}/\sqrt{r}, (\sum_{j=0}^{p} \lambda_{i_j)} + \varepsilon^{'}/\sqrt{r})$. Let us show that this will hold whenever each $\lambda_{i_j}^{'} \in (\lambda_{i_j} - \varepsilon^{'}/n\sqrt{r}, \lambda_{i_j} + \varepsilon^{'}/n\sqrt{r})$. Assume that every $|\lambda_{i_j}^{'} - \lambda_{i_j}| < \varepsilon^{'}/n\sqrt{r}$. Then $|\sum_{j=0}^{p} \lambda_{i_j}^{'} - \sum_{j=0}^{p} \lambda_{i_j}| \leq \sum_{j=0}^{p} |\lambda_{i_j}^{'} - \lambda_{i_j}| < \sum_{j=0}^{p} \varepsilon^{'}/n\sqrt{r} = p\varepsilon^{'}/n\sqrt{r} < \varepsilon^{'}/\sqrt{r}$, since $p < n$.

Thus, take $\delta = \varepsilon^{'}/n\sqrt{r}$. Assume that $\rho(x,z) < \delta$. From this follows that for all coordinates of $\lambda_i^{'}$ of z we have that $\lambda_i^{'} \in (\lambda_i - \delta, \lambda_i + \delta)$. Then, by the previous reasoning, we obtain that $\mu_i^{'} \in (\mu_i - \varepsilon^{'}/\sqrt{r}, \mu_i + \varepsilon^{'}/\sqrt{r})$. Therefore, $\rho(f^*(x), f^*(z)) < \varepsilon$.

We conclude that f^* is continuous. $\square$

Definition 5.4.31. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic monotone frame. We will call $\mathcal{D} = (|\mathcal{K}|, \mathcal{K}, f^*)$ the dynamic realisation of $\mathcal{F}^d$ if:

1. $\mathcal{K}$ is a simplicial complex of $\mathcal{F}^d$;
2. f^* is the dynamic protagonist of $\mathcal{F}^d$.

Lemma 5.4.32. Let $\mathcal{F}^d = (W, \sqsubseteq, R)$ be a dynamic monotone frame and $\mathcal{D} = (|\mathcal{K}|, \mathcal{K}, f^*)$ its dynamic realisation.

Proof. Follows from Lemma 5.4.30 and Lemma 5.4.29. $\square$

Thus, we have that $\mathcal{D} = (|\mathcal{K}|, \mathcal{K}, f^*)$ is a dynamic polyhedral system, and by Lemma 5.4.26, we obtain that $(\mathcal{K}, \precsim, \tilde{f})$ is its encoding.

Theorem 5.4.33. For a dynamic monotone frame $\mathcal{F}^d = (W, \sqsubseteq, f)$ and its dynamic realisation $\mathcal{D} = (|\mathcal{K}|, \mathcal{K}, f^*)$:

$$\mathcal{D} \vDash \varphi \Rightarrow \mathcal{F}^d \vDash \varphi,$$

for every φ in $\mathcal{L}_{\square\langle\mathrm{R}\rangle}$.

Proof. Analogous to theorem 5.4.21. $\square$

We finally arrive at the main theoretical result of the thesis.

Theorem 5.4.34.

1. Logic $\mathcal{DRL}$ is complete with respect to the class of all dynamic relational systems.
2. Logic $\mathcal{DPL}$ is complete with respect to the class of all dynamic polyhedral systems.

Proof. We will present the proof only for $\mathcal{DPL}$ since the proof for $\mathcal{DRL}$ is completely analogous. Assume that $\varphi \notin \mathcal{DPL}$. Then there is a dynamic monotone frame $\mathcal{F}^d$ such that $\mathcal{F}^d \nvDash \varphi$. Then, taking the dynamic realization of $\mathcal{F}^d$, $\mathcal{D}$ which corresponds to it, we obtain that $\mathcal{D} \nvDash \varphi$ by Theorem 5.4.33. $\square$

We finish this chapter with a summary of what has been done.

- We defined two logics $\mathcal{DRL}$ and $\mathcal{DPL}$, dynamic relational logic and dynamic polyhedral logic respectively.

- We proved that $\mathcal{DRL}$ is valid on class of dynamic relational systems ($\mathbf{DRS}$) and that $\mathcal{DPL}$ is valid the class of dynamic polyhedral systems ($\mathbf{DPS}$).
- We proved the completeness of $\mathcal{DRL}$ and $\mathcal{DPL}$ with respect to the class of finite naïvely monotone dynamic Kripke frames and finite monotone dynamic Kripke frames, respectively.
- Using the method of constructing a polyhedron from a finite Kripke frame defined in [Ada19], we described how to build a dynamic relational system $\mathcal{D}$ from a finite naïvely monotone dynamic Kripke frame $\mathcal{F}^d$ such that if $\mathcal{D} \vDash \varphi$, then $\mathcal{F}^d \vDash \varphi$ for every φ in language $\mathcal{L}_{\Box,\langle\mathrm{R}\rangle}$. The same result was obtained for dynamic polyhedral and finite monotone dynamic systems.

Chapter 6

Conclusion and future work

In this thesis, we have presented a novel approach to the specification and automatic verification of dynamic polyhedral structures' properties. The required background on the static semantics is presented in Chapter 2, together with a form of a bisimilarity for finite Kripke frames and the proof of the Hennessy-Milner property.

Then in Chapter 3 we defined two novel forms of dynamic models: dynamic polyhedral models (Definition 3.1.9) and dynamic relational models (Definition 3.3.3). We demonstrated how to create finite Kripke frames as encodings for these models. We also proved that in order to verify if a formula is true at a specific point of a dynamic polyhedral model or a dynamic relational model $\mathcal{O}$, one only needs to check whether the formula is true in the corresponding point of the encoding $\mathcal{M}(\mathcal{O})$ (Theorems 3.2.11 and 3.3.14).

In Chapter 4, we introduced a model checking algorithm for verifying the truth of a formula on dynamic polyhedral models and dynamic relational models, utilizing their encodings. Additionally, we demonstrated the application of the existing model checker `PolyLogicA` to a new case study. We have also modified the source code of the tool by describing the steps required to apply it to a real 3D model scenario. Based on this example, we outlined a conceptual design for a prototype of a dynamic model checker.

In Chapter 5 we introduced the logics $\mathcal{DPL}$ and $\mathcal{DRL}$ and showed that they enjoy the finite model property with respect to Kripke frames and are complete with respect to dynamic polyhedral semantics.

There are still many problems that remain open. For example, what results can we get if we change the language? It is natural to add the modality "forever in the future" ($[\forall]$). See Figure 6.1 for all the possible variants of the languages. In our study, we addressed the question of completeness of logics in the language $\mathcal{L}_{\Box,\langle R\rangle}$, and also defined model checking algorithm for models and formulas in the language

$$
\begin{tikzcd}
& & \mathcal{L}_{\langle \mathrm{R} \rangle,\gamma,\forall} & & \\
\mathcal{L}_{\Box,\langle \mathrm{R} \rangle,[\forall]} \arrow[urr] & & \mathcal{L}_{\langle \mathrm{R} \rangle,\gamma} \arrow[u] & & \mathcal{L}_{\gamma,[\forall]} \arrow[ull] \\
& \mathcal{L}_{\Box,\langle \mathrm{R} \rangle} \arrow[ul] \arrow[ur] & & \mathcal{L}_{\Box,\gamma} \arrow[ul] \arrow[ur] & \\
& & \mathcal{L}_{\Box} \arrow[ul] \arrow[ur] & &
\end{tikzcd}
$$

Figure 6.1: Dynamic polyhedral languages

$\langle \mathrm{R} \rangle, \gamma$. The same question can be posed for languages in Figure 6.1. Let us consider some of the most interesting ones.

1. $\mathcal{L}_{\Box,\gamma}$ – The question of the axiomatization of the minimal logic in this language remains open. After the axiomatization result, it is quite natural to ask about the completeness of defined logic with respect to all polyhedra;

2. $\mathcal{L}_{\langle \mathrm{R} \rangle,\gamma}$ – the questions of axiomatization and completeness for logics in this language would also be a very non-trivial challenge, which could be the first step for understanding the relation between γ and the dynamics;

3. $\mathcal{L}_{\Box,\langle \mathrm{R} \rangle,[\forall]}$ – extension of the language with the modality $[\forall]$ is a standard feature in dynamic logic [KM07]. The issues of axiomatization, completeness, and definition of model checking algorithm would be very interesting problems to solve.

4. $\mathcal{L}_{\langle \mathrm{R} \rangle,\gamma,[\forall]}$ – this language is very expressive, the questions axiomatization and completeness for it may require a lot of effort, but still would be non-trivial results. The main problem with this language is that it is not apparent how the two fixpoint operators γ and $[\forall]$ should be related to each other. A more feasible result for this logic would be a design of a model checking algorithm for formulas in this language.

Finally, a practical challenge that remains unsolved is a development of a prototype for model checking in dynamic 3D models. To tackle this, it seems more promising to adopt the $\mathcal{L}_{\langle \mathrm{R} \rangle,\gamma,[\forall]}$ language instead of $\mathcal{L}_{\langle \mathrm{R} \rangle,\gamma}$, as it offers greater expressiveness. Consequently, the initial phase of this project requires a formulation of a model checking algorithm. The construction of the prototype requires considerable software development skills and an understanding of the specific requirements of the image analysis domain.

Bibliography

[Ada+22] Sam Adam-Day et al. “Polyhedral completeness of intermediate logics: the Nerve Criterion”. In: *The Journal of Symbolic Logic* (2022), pp. 1–41.

[Ada19] Sam Adam-Day. “Polyhedral completeness in intermediate and modal logics”. In: (2019).

[ADN97] Sergei Artemov, Jennifer Davoren, and Anil Nerode. *Modal logics and topological semantics for hybrid systems*. Tech. rep. Technical Report MSI 97-05, Cornell University, 1997.

[BB07] Johan van Benthem and Guram Bezhanishvili. “Modal logics of space”. In: *Handbook of spatial logics*. Springer, 2007, pp. 217–298.

[BBF21] Alexandru Baltag, Nick Bezhanishvili, and David Fernández-Duque. “The topological mu-calculus: completeness and decidability”. In: *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. IEEE. 2021, pp. 1–13.

[BBH17] Johan van Benthem, Nick Bezhanishvili, and Wesley H Holliday. “A bimodal perspective on possibility semantics”. In: *Journal of Logic and Computation* 27.5 (2017), pp. 1353–1389.

[BDV01] Patrick Blackburn, Maarten De Rijke, and Yde Venema. *Modal logic: graph. Darst*. Vol. 53. Cambridge University Press, 2001.

[Bez+18] Nick Bezhanishvili et al. “Tarski’s theorem on intuitionistic logic, for polyhedra”. In: *Annals of Pure and Applied Logic* 169.5 (2018), pp. 373–391.

[Bez+22] Nick Bezhanishvili et al. “Geometric model checking of continuous space”. In: *Logical Methods in Computer Science* 18 (2022).

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

[Cha97] Alexander Chagrov. *Modal Logic*. Ed. by Michael Zakharyaschev. New York: Oxford University Press, 1997.

[Cia+14] Vincenzo Ciancia et al. "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 8*. Springer. 2014, pp. 222–235.

[Cia+23] Vincenzo Ciancia et al. "Minimisation of Spatial Models Using Branching Bisimilarity". In: *Formal Methods*. Ed. by Marsha Chechik, Joost-Pieter Katoen, and Martin Leucker. Cham: Springer International Publishing, 2023, pp. 263–281. ISBN: 978-3-031-27481-7.

[Cig+08] Paolo Cignoni et al. "MeshLab: an Open-Source Mesh Processing Tool". In: *Eurographics Italian Chapter Conference*. Ed. by Vittorio Scarano, Rosario De Chiara, and Ugo Erra. The Eurographics Association, 2008. ISBN: 978-3-905673-68-5. DOI: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.

[FM21] David Fernández-Duque and Yoàv Montacute. "Dynamic Cantor Derivative Logic". In: *arXiv preprint arXiv:2107.10349* (2021).

[GH97] Michael Garland and Paul S Heckbert. "Surface simplification using quadric error metrics". In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. 1997, pp. 209–216.

[Jec03] Thomas Jech. *Set theory: The third millennium edition, revised and expanded*. Springer, 2003.

[KM07] Philip Kremer and Grigori Mints. "Dynamic Topological Logic". In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian Pratt-Hartmann, and Johan Van Benthem. Dordrecht: Springer Netherlands, 2007, pp. 565–606. ISBN: 978-1-4020-5587-4. DOI: 10.1007/978-1-4020-5587-4_10. URL: https://doi.org/10.1007/978-1-4020-5587-4_10.

[Kon+07] Roman Kontchakov et al. "Spatial Logic + Temporal Logic = ?" In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian Pratt-Hartmann, and Johan Van Benthem. Dordrecht: Springer Netherlands, 2007, pp. 497–564. ISBN: 978-1-4020-5587-4. DOI: 10.1007/978-1-4020-5587-4_9. URL: https://doi.org/10.1007/978-1-4020-5587-4_9.

[MT44] John Charles Chenoweth McKinsey and Alfred Tarski. "The algebra of topology". In: *Annals of mathematics* (1944), pp. 141–191.

[RS12] Colin P Rourke and Brian Joseph Sanderson. *Introduction to piecewise-linear topology*. Springer Science & Business Media, 2012.

[She99] Valentin Shehtman. “Everywhere» and «here”. In: *Journal of Applied Non-Classical Logics* 9.2-3 (1999), pp. 369–379.

[Tar39] Alfred Tarski. “Der aussagenkalkul und die topologie”. In: *Journal of Symbolic Logic* 4.1 (1939).