

# Finitely-valued Propositional Dynamic Logic

Igor Sedlár<sup>1</sup>

*The Czech Academy of Sciences, Institute of Computer Science  
Pod Vodárenskou věží 271/2  
Prague, The Czech Republic*

---

## Abstract

We study a many-valued generalization of Propositional Dynamic Logic where formulas in states and accessibility relations between states of a Kripke model are evaluated in a finite FL-algebra. One natural interpretation of this framework is related to reasoning about costs of performing structured actions. We prove that PDL over any finite FL-algebra is decidable. We also establish a general completeness result for a class of PDLs based on commutative integral FL-algebras with canonical constants.

*Keywords:* FL-algebras, Many-valued modal logic, Propositional Dynamic Logic, Residuated lattices, Substructural logics, Weighted structures.

---

## 1 Introduction

Propositional dynamic logic, PDL, is a well-known modal logic formalizing reasoning about structured actions, e.g. computer programs or actions performed by physical agents, and their correctness properties [10,17]. PDL is subject to two limiting design features. First, being based on classical propositional logic, it formalizes actions that modify values of Boolean variables. A more general setting, one where variables take values from an arbitrary set (integers, characters, trees etc.), is offered by variants of first-order Dynamic Logic, DL [16,17]; these variants, however, are mostly undecidable. Second, PDL can express the fact that one action is guaranteed to attain a certain goal while another action is not, but it is not able to express that one action is a more efficient way of attaining the goal than another action. In other words, accessibility between states mediated by actions is modelled as a crisp rather than a graded relation; the former approach is a convenient idealization, but the latter one is more realistic and often also practically required.

---

<sup>1</sup> Email: sedlar@cs.cas.cz. This work was supported by the Czech Science Foundation grant GJ18-19162Y for the project *Non-Classical Logical Models of Information Dynamics*. Some results presented here were obtained while the author was visiting Manuel A. Martins and Alexandre Madeira at the University of Aveiro; our discussions helped to initiate this work and shaped the paper. The author is also grateful to Petr Cintula and Libor Běhounek for discussions on their earlier work on many-valued dynamic logic and the audience of the Seminar of Applied Mathematical Logic at the Institute of Computer Science CAS. The valuable comments of the anonymous reviewers are gratefully acknowledged.

Both of these limitations of “classical” PDL are avoided in a *many-valued* setting. In such a setting, values of formulas in states of a Kripke model are taken from an algebra that is typically distinct from the two-element Boolean algebra used in classical PDL. In a many-valued setting, accessibility between states can also be evaluated in such an algebra, naturally leading to a representation of “costs” or other “weights” associated with performing actions under specific circumstances.

Research into many-valued modal logics dates back to the 1960s, see the pioneering [25] and the later [23]. Fitting [11,12] was the first to study modal logics where both formulas in states and accessibility relations between states in the Kripke model take values from a non-Boolean algebra. Fitting considers finite Heyting algebras; generalizations studied for example in [4,7,6,15,28] focus on various kinds of finite or infinite *residuated lattices* [13]. Residuated lattices are algebraic structures related to *substructural logics*, with many important special cases such as Boolean and Heyting algebras, relation algebras, lattice-ordered groups, powersets of monoids, various algebras on the  $[0, 1]$ -interval and so on.

Investigations of PDL based on residuated lattices are relatively scarce. The work in [5,18,19] focuses on expressivity of PDL with many-valued accessibility, but technical results such as decidability or completeness are not provided. Teheux [27] establishes decidability and completeness of PDLs based on finite Lukasiewicz chains and the present author [24] establishes decidability and completeness of PDL extending the paraconsistent modal logic of [22]; both papers, however, deal with crisp accessibility relations. As an attempt to systematize the work in many-valued PDL, Madeira et al. [20,21] put forward a general method of producing many-valued versions of PDL, based on the matrix representation of Kleene algebras; their method, however, applies only if models are defined to be finite.

In this paper we add to this literature by studying PDLs based on finite Full Lambek algebras, that is, residuated lattices with a distinguished, though arbitrary, 0 element. We assume that both evaluations of formulas in states and accessibility between states are many-valued. Our main technical results are general completeness and decidability proofs for logics in the family. To the best of our knowledge, our results are the first decidability and completeness results concerning non-crisp many-valued PDL. To be more specific, we work with versions of test-free PDL based on finite Full Lambek algebras with canonical constants; we prove that any PDL based on a finite FL-algebra with canonical constants is decidable; we also establish a completeness result for PDLs based on finite commutative integral FL-algebras with canonical constants.

The paper is structured as follows. Section 2 introduces the general framework of PDL based on finite FL-algebras. We note that, for technical reasons discussed in §6, our version of PDL uses the transitive closure operator, or Kleene plus, as primitive instead of the more standard reflexive transitive closure operator, the Kleene star. An informal interpretation of the framework is discussed in §3. Section 4 establishes our decidability result using a generaliza-

tion of the smallest filtration technique. Section 5 establishes the completeness result for PDLs based on finite integral commutative FL-algebras with canonical constants. Our work there builds on the results of [4], but the canonical model construction used in our proof is novel to this paper (it is a suitable generalization of the greatest filtration construction, though the model itself is infinite).

## 2 Preliminaries

In this section we briefly recall two-valued PDL (§2.1), and we define FL-algebras and many-valued models for the language of PDL based on them (§2.2). We point out some basic facts that we will use later on.

### 2.1 Two-valued PDL

We begin by recalling some well-known facts about two-valued test-free PDL; see [17]. Fix  $Ac = \{a_i \mid i \in \omega\}$ , a countable set of atomic action expressions. The set of *standard action expressions*,  $STA$ , is the closure of  $Ac$  under applying binary operators  $;$  (“composition”),  $\cup$  (“choice”) and unary  $*$  (“Kleene star”). That is,  $STA$  are regular expressions over  $Ac$  without the empty expression. For example,  $(a_0; a_1)^* \cup a_0$  is in  $STA$ . Let  $Pr = \{p_i \mid i \in \omega\}$  be a countable set of propositional variables. Take  $\mathbf{2}$ , the two-element Boolean algebra on the set  $\{0, 1\}$  with meet  $\sqcap$ , join  $\sqcup$  and complement  $-$ ; the binary operation  $\Rightarrow$  is defined as usual:  $a \Rightarrow b := -a \sqcup b$ . Formulas of the *standard language for  $\mathbf{2}$* ,  $Fm(\mathcal{L}_2^{STA})$ , are defined by

$$\varphi := p \mid \bar{c} \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid [\alpha]\varphi$$

where  $p \in Pr$ ,  $c \in \mathbf{2}$  and  $\alpha \in STA$ . For example,  $p_0 \rightarrow [a_0; (a_1)^*](p_1 \rightarrow \bar{0})$  is a formula of  $\mathcal{L}_2^{STA}$ .

A  *$\mathbf{2}$ -valued frame for  $STA$*  is  $\mathfrak{F} = (S, \{R_\alpha\}_{\alpha \in STA})$  where  $S$  is a non-empty set and, for each  $\alpha \in STA$ ,  $R_\alpha$  is a function from  $S \times S$  to  $\mathbf{2}$ . We denote  $R(\alpha) := \{(s, t) \mid R_\alpha(s, t) = 1\}$ ; and the functions in  $\{R_\alpha\}_{\alpha \in STA}$  are required to satisfy the following: (i)  $R(\alpha \cup \beta) = R(\alpha) \cup R(\beta)$ ; (ii)  $R(\alpha; \beta) = R(\alpha) \circ R(\beta)$ , the composition of  $R(\alpha)$  and  $R(\beta)$ ; (iii)  $R(\alpha^*) = R(\alpha)^*$ , the reflexive transitive closure of  $R(\alpha)$ .

Let  $\mathfrak{F} = (S, \{R_\alpha\}_{\alpha \in STA})$  be a  $\mathbf{2}$ -valued frame. A  *$\mathbf{2}$ -valued model based on  $\mathfrak{F}$*  is  $\mathfrak{M} = (S, \{R_\alpha\}_{\alpha \in STA}, V)$  where  $V : Fm(\mathcal{L}_2^{STA}) \times S \rightarrow \mathbf{2}$  such that

- $V(\bar{c}, s) = c$ ;
- $V(\varphi \wedge \psi, s) = V(\varphi, s) \sqcap V(\psi, s)$ ,  $V(\varphi \vee \psi, s) = V(\varphi, s) \sqcup V(\psi, s)$ , and  $V(\varphi \rightarrow \psi, s) = V(\varphi, s) \Rightarrow V(\psi, s)$ ;
- $V([\alpha]\varphi, s) = \prod_{t \in S} (R_\alpha(s, t) \Rightarrow V(\varphi, t))$ .

Note that  $V([\alpha]\varphi, s) = \prod_{R_\alpha(s, t)=1} V(\varphi, t)$ . A formula  $\varphi$  is *valid in  $\mathfrak{M}$*  iff  $V(\varphi, s) = 1$  for all  $s$ ; validity in frames and classes of frames is defined as expected.

This is the standard presentation of test-free PDL, phrased in a way that invites generalizations obtained by replacing  $\mathbf{2}$  by another algebra. We will

study some such generalizations in this paper but, as we discuss in more detail below, the story is somewhat more complicated. For reasons discussed in §6, our generalizations will use a different primitive iteration operator instead of the Kleene star. The operator we will use, however, is conveniently related to the Kleene star.

The set of *action expressions over  $Ac$* ,  $ACT$ , is the closure of  $Ac$  under composition, choice and the unary operator  $+$  (“Kleene plus”). Formulas of the *language  $\mathcal{L}_2$*  are defined as expected (we omit reference to  $ACT$ ), with  $\alpha \in ACT$ ; for example,  $p_0 \rightarrow [a_0; (a_1)^+](p_1 \rightarrow \bar{0})$  is a formula of  $\mathcal{L}_2$ . The definition of *2-valued frames for  $ACT$*  is the same as the definition of **2**-valued frames for  $STA$ , with an obvious exception, namely, the requirement that  $R(\alpha^+)$  be the *transitive closure* of  $R(\alpha)$ , i.e.  $R(\alpha^+) = \bigcup_{n>0} R^n(\alpha)$ , where  $R^1(\alpha) = R(\alpha)$  and  $R^{n+1}(\alpha) = R^n(\alpha) \circ R(\alpha)$ . Compare this with the *reflexive transitive closure*  $R(\alpha)^* = \bigcup_{n \geq 0} R^n(\alpha)$ , where  $R^0(\alpha) = \{(s, s) \mid s \in S\}$ . Models based on frames for  $ACT$  are defined as before.

**Proposition 2.1** *Let For each  $\alpha \in ACT$  and  $\varphi \in Fm(\mathcal{L}_2)$ ,*

$$V(\varphi \wedge [\alpha^+] \varphi, s) = 1 \quad \text{iff} \quad \forall t((s, t) \in R(\alpha)^* \implies V(\varphi, t) = 1).$$

Proposition 2.1 implies that  $\varphi \wedge [\alpha^+] \varphi$  “simulates”  $[\alpha^*] \varphi$  in  $\mathcal{L}_2$ . This provides a justification for our using languages based on  $ACT$  rather than on  $STA$  in what follows. However, we admit that this choice is related to the technical issues discussed in §6.

## 2.2 FL-algebras and finitely-valued PDL

In this section we generalize two-valued PDL by replacing the two-element Boolean algebra **2** by a more general structure, namely, a finite FL-algebra. FL-algebras provide semantics for a wide class of *substructural logics* [13].

**Definition 2.2** An *FL-algebra* (“full Lambek algebra”, [13]) is a set  $X$  with binary operations  $\sqcap, \sqcup, \backslash, \cdot, /$  and two distinguished elements  $1, 0$  such that

- $(X, \sqcap, \sqcup)$  is a lattice (let  $a \sqsubseteq b$  iff  $a \sqcup b = b$ );
- $(X, \cdot, 1)$  is a monoid;
- $(\backslash, \cdot, /)$  are residuated over  $(X, \sqsubseteq)$ , i.e.

$$a \cdot b \sqsubseteq c \quad \text{iff} \quad b \sqsubseteq a \backslash c \quad \text{iff} \quad a \sqsubseteq c / b;$$

- $0$  is an arbitrary element of  $X$ .

*Residuated lattices* are 0-free reducts of FL-algebras. Each finite FL-algebra  $\mathbf{X}$  contains a least element  $\perp^{\mathbf{X}}$  (for all  $a \in X$ ,  $\perp^{\mathbf{X}} \sqsubseteq a$ ) and a greatest element  $\top^{\mathbf{X}}$  (for all  $a \in X$ ,  $a \sqsubseteq \top^{\mathbf{X}}$ ).

We usually write  $ab$  instead of  $a \cdot b$  and  $a \Rightarrow b$  instead of  $b/a$ . Two varieties of FL-algebras will be important in this paper:

- *commutative* FL-algebras satisfy  $ab = ba$  for all  $a, b \in X$ ;

- *integral* FL-algebras satisfy  $a \sqsubseteq 1$  for all  $a \in X$ .

Note that in commutative FL-algebras  $a \setminus b = b/a$ .

**Example 2.3** The two-element Boolean algebra **2** is a commutative integral FL-algebra, where  $\cdot$  is  $\sqcap$  and  $\setminus$  (identical to  $/$ ) is  $\Rightarrow$ .

**Example 2.4** Let  $N > 0$  and define  $\mathbf{N} = (N, \max, \min, +_N, \rightarrow_N)$  where

$$a +_N b = \min(a + b, N - 1) \quad \text{and} \quad a \rightarrow_N b = \max(b - a, 0).$$

$\mathbf{N}$  is a finite commutative integral FL-algebra, with 0 as the monoid identity with respect to  $+_N$  and the greatest element under the  $\geq$ -ordering induced by taking  $\min$  as join. We note that  $\mathbf{N}$  is isomorphic to the  $N$ -element Łukasiewicz lattice  $\mathbf{L}_N$  over  $\{\frac{k}{N-1} \mid k \in N\}$ .

**Example 2.5** As an example of a non-commutative, non-integral infinite FL-algebra, take the power set of the free monoid over some set  $\Sigma$ , i.e. the set of languages over  $\Sigma$ , with intersection as meet, union as join,  $L \cdot L' := \{xx' \mid x \in L \ \& \ x' \in L'\}$ ,  $\{\varepsilon\}$  as the monoid identity ( $\varepsilon$  is the empty word) and  $L \setminus L' := \{x \in \Sigma \mid L \cdot \{x\} \subseteq L'\}$ ,  $L'/L := \{x \in \Sigma \mid \{x\} \cdot L \subseteq L'\}$ .

The following lemma summarizes some of the properties of FL-algebras we will rely on in this paper (we will often say that something holds “by the properties FL-algebras” in our proofs).

**Lemma 2.6** *Let  $\mathbf{X}$  be an arbitrary FL-algebra. Then* (i)  $a \sqsubseteq b$  iff  $1 \sqsubseteq a \Rightarrow b$ ; (ii) If  $a \sqsubseteq b$  and  $c \sqsubseteq d$ , then  $b \Rightarrow c \sqsubseteq a \Rightarrow d$ ,  $b \setminus c \sqsubseteq a \setminus d$  and  $ac \sqsubseteq bd$ ; (iii)  $(a \sqcup b)c = ac \sqcup ab$  and  $c(a \sqcup b) = ca \sqcup cb$ ; (iv)  $a \Rightarrow (b \sqcap c) = (a \Rightarrow b) \sqcap (a \Rightarrow c)$ ; (v)  $a \sqcup b \Rightarrow c = (a \Rightarrow c) \sqcap (b \Rightarrow c)$ ; (vi)  $a \Rightarrow (b \Rightarrow c) = ab \Rightarrow c$ ; (vii)  $(a \Rightarrow b)(b \Rightarrow c) \sqsubseteq a \Rightarrow c$ ; (viii)  $(1 \Rightarrow a) = a$

If  $S$  is a non-empty set, then  $\Pi(S)$  is the set of all finite sequences of elements of  $S$ ; that is,  $\pi \in \Pi(S)$  iff  $\pi$  is a function from some  $n \in \omega$ , called the length of  $\pi$ , to  $S$ . The unique sequence of length 0 is  $\emptyset$ . If  $\pi$  is a sequence of length  $n$  and  $s \in S$ , then  $\pi \hat{\ } s$  is the unique sequence of length  $n + 1$  such that  $(\pi \hat{\ } s)(k) = \pi(k)$  for all  $k < n$  and  $(\pi \hat{\ } s)(n) = s$ . Note that each sequence  $\pi$  of length  $n > 0$  can be expressed as  $(\dots (\emptyset \hat{\ } \pi(0)) \hat{\ } \dots) \hat{\ } \pi(n - 1)$ .

**Definition 2.7** Let  $\mathbf{X}$  be a finite FL-algebra and  $S$  a non-empty set. A *binary  $\mathbf{X}$ -valued relation on  $S$*  is any function from  $S \times S$  to  $\mathbf{X}$ . Let  $R, Q$  be binary  $\mathbf{X}$ -valued relations on a set  $S$ ; then

- the *union* of  $R$  and  $Q$  is the function  $R \cup Q$  defined by  $(R \cup Q)(s, t) := R(s, t) \sqcup Q(s, t)$ ;
- the *composition* of  $R$  and  $Q$  is the function  $R \circ Q$  defined by  $(R \circ Q)(s, t) = \bigsqcup_{x \in S} (R(s, x) \cdot Q(x, t))$ ;
- the *transitive closure* of  $R$  is the function  $R^+$  defined by  $R^+(s, t) = \bigsqcup_{\pi \in \Pi(S)} R s \pi t$  where  $R s \pi t$  is defined as follows:
  - $R s \emptyset t = R(s, t)$  and
  - $R s (\pi \hat{\ } u) t = R s \pi u \cdot R(u, t)$ .

We say that  $Q$  extends  $R$ , notation  $R \sqsubseteq Q$ , iff  $R(s, t) \sqsubseteq Q(s, t)$  for all  $s, t \in S$ ;  $R$  is the *smallest* relation in a set  $\{R_i\}_{i \in I}$  if  $R = R_i$  for some  $i \in I$  and each  $R_i$  extends  $R$ .  $R$  is *transitive* if  $R(s, t) \cdot R(t, u) \sqsubseteq R(s, u)$  for all  $s, t, u \in S$ ; and  $R$  is *reflexive* if  $1 \sqsubseteq R(s, s)$  for all  $s \in S$ .

Note that we need to assume that all the required joins exist in  $\mathbf{X}$ ; hence the restriction to finite FL-algebras (however, a restriction to complete  $\mathbf{X}$  is sufficient, as is the assumption that  $R, Q$  are “ $\mathbf{X}$ -safe” [14, ch. 5]).

**Proposition 2.8** *Let  $\mathbf{X}$  be a finite FL-algebra and  $R$  a binary  $\mathbf{X}$ -valued relation on a set  $S$ . Then  $R^+$  is the smallest transitive relation extending  $R$ . For any  $R$ , define  $R^*$  as follows:*

$$R^*(s, t) = \begin{cases} 1 & \text{if } s = t \\ R^+(s, t) & \text{otherwise.} \end{cases}$$

*Then  $R^*$  is the smallest reflexive transitive relation extending  $R$ .*

**Proof.** It is clear that  $R^+$  is a transitive relation extending  $R$ . Now assume that so is  $Q$ . The conclusion that  $R^+ \sqsubseteq Q$  follows from two facts that are easily established by induction on the length of  $\pi$ : (a) For all  $s, t \in S$  and  $\pi \in \Pi(S)$ ,  $R s \pi t \sqsubseteq Q s \pi t$  (the assumption that  $R \sqsubseteq Q$  is used here); (b) For all  $s, t \in S$  and  $\pi \in \Pi(S)$ ,  $Q s \pi t \sqsubseteq Q(s, t)$  (the assumption that  $Q$  is transitive is used). Since  $\mathbf{X}$  is finite, the two claims imply that, for any given  $s$  and  $t$ ,  $\bigsqcup_{\pi} R s \pi t \sqsubseteq \bigsqcup_{\pi} Q s \pi t \sqsubseteq Q(s, t)$ .

It is clear that  $R^*$  is a reflexive transitive relation extending  $R$ . If so is  $Q$ , then we reason for any given  $s$  and  $t$  by cases as follows. If  $s = t$ , then  $R^*(s, t) \sqsubseteq Q(s, t)$  is equivalent to  $1 \sqsubseteq Q(s, s)$ , which holds by reflexivity of  $Q$ . If  $s \neq t$ , then  $R^*(s, t) \sqsubseteq Q(s, t)$  is equivalent to  $R^+(s, t) \sqsubseteq Q(s, t)$ , which follows from the assumption that  $Q$  is a transitive relation extending  $R$ . Hence,  $R^*(s, t) \sqsubseteq Q(s, t)$  for any  $s$  and  $t$ .  $\square$

**Lemma 2.9** *Let  $\mathbf{X}$  be a finite FL-algebra and  $S$  a set; the  $\mathbf{X}$ -valued identity relation on  $S$  is defined as follows:*

$$Id_{\mathbf{X}}(s, t) := \begin{cases} 1 & \text{if } s = t \\ \perp^{\mathbf{X}} & \text{otherwise.} \end{cases}$$

*If  $\mathbf{X}$  is integral, then  $R^* = Id_{\mathbf{X}} \cup R^+$  for any binary  $\mathbf{X}$ -valued relation on  $S$ .*

**Proof.** We omit the proof; we just note that if  $s = t$ , then  $R^*(s, t) = Id_{\mathbf{X}}(s, t) \sqcup R^+(s, t)$  is equivalent to  $R^+(s, t) \sqsubseteq 1$ , which is guaranteed to hold only if  $\mathbf{X}$  is integral.  $\square$

**Definition 2.10** Let  $\mathbf{X}$  be a finite FL-algebra. An  $\mathbf{X}$ -valued frame for ACT is a pair  $\mathfrak{F} = (S, \{R_{\alpha}\}_{\alpha \in ACT})$  where  $S$  is a non-empty set and, for all  $\alpha \in ACT$ ,  $R_{\alpha}$  is an  $\mathbf{X}$ -valued binary relation on  $S$  such that (i)  $R_{\alpha \cup \beta} = R_{\alpha} \cup R_{\beta}$ ; (ii)  $R_{\alpha; \beta} = R_{\alpha} \circ R_{\beta}$ ; and (iii)  $R_{\alpha^+} = R_{\alpha}^+$ .

$\mathbf{X}$ -valued frames will also be referred to as  $\mathbf{X}$ -frames or simply frames if  $\mathbf{X}$  is clear from the context or immaterial. We will sometimes write  $R_\alpha st$  instead of  $R_\alpha(s, t)$ .

**Definition 2.11** Formulas of the language  $\mathcal{L}_\mathbf{X}$  are defined as follows:

$$\varphi := p \mid \bar{c} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \setminus \varphi \mid \varphi \cdot \varphi \mid \varphi / \varphi \mid [\alpha]\varphi,$$

where  $p \in Pr$ ,  $c \in \mathbf{X}$  and  $\alpha \in ACT$ . We use  $\perp, \top$  instead of  $\overline{\perp^\mathbf{X}}$  and  $\overline{\top^\mathbf{X}}$ , respectively. We often write  $\varphi\psi$  instead of  $\varphi \cdot \psi$ ,  $\varphi \rightarrow \psi$  instead of  $\psi / \varphi$ ,  $m$  instead of  $a_m$ , and  $\alpha\beta$  instead of  $\alpha; \beta$ . We define  $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$ ,  $\neg\varphi := \varphi \rightarrow \perp$  and  $\langle \alpha \rangle \varphi := \neg[\alpha]\neg\varphi$ .

Note that we use the same symbol  $\otimes \in \{\setminus, \cdot, /\}$  for the implication and fusion connectives of the language and for the residuated operations on FL-algebras. We will denote the operations on a given  $\mathbf{X}$  as  $\otimes^\mathbf{X}$  in contexts where it is convenient for the reader to distinguish the connectives of the language from the operations on the algebra. (However,  $\Rightarrow$  denotes the operation  $/^\mathbf{X}$  and  $\rightarrow$  denotes the connective  $/$  throughout.)

**Definition 2.12** A model based on an  $\mathbf{X}$ -frame  $(S, \{R_\alpha\}_{\alpha \in ACT})$  is  $\mathfrak{M} = (S, \{R_\alpha\}_{\alpha \in ACT}, V)$ , where  $V$  is a function from  $Fm(\mathcal{L}_\mathbf{X}) \times S$  to  $\mathbf{X}$  such that

- $V(\bar{c}, s) = c$ ;
- $V(\varphi \wedge \psi, s) = V(\varphi, s) \sqcap V(\psi, s)$  and  $V(\varphi \vee \psi, s) = V(\varphi, s) \sqcup V(\psi, s)$ ;
- $V(\varphi \otimes \psi, s) = V(\varphi, s) \otimes^\mathbf{X} V(\psi, s)$  for  $\otimes \in \{\setminus, \cdot, /\}$ ;
- $V([\alpha]\varphi, s) = \prod_{t \in S} (R_\alpha st \Rightarrow V(\varphi, t))$ .

A formula  $\varphi$  is *valid in  $\mathfrak{M}$*  iff  $1 \sqsubseteq V(\varphi, s)$  for all  $s$  in  $\mathfrak{M}$ . Validity in frames and classes of frames is defined as expected. The *theory* of a frame is the set of formulas valid in the frame; the theory of a class of frames is the set of formulas valid in each frame in the class.  $Th(\mathbf{X})$  is the theory of the class of all  $\mathbf{X}$ -frames.

The following addendum to Proposition 2.1 suggests that integral FL-algebras are particularly suitable for us.

**Proposition 2.13** Take an arbitrary  $\mathbf{X}$ -frame for a finite integral  $\mathbf{X}$ . Then  $V(\varphi \wedge [\alpha^+]\varphi, s) = \prod_{t \in S} (R_\alpha^* st \Rightarrow V(\varphi, t))$ .

**Proof.** The  $\sqsubseteq$ -inequality is straightforward and the  $\sqsupseteq$ -inequality follows from Lemma 2.9. □

It is clear that two-valued PDL is a special case of the present framework for  $\mathbf{X} = \mathbf{2}$ .

**Lemma 2.14** The following are valid in each  $\mathbf{X}$ -frame:

$$\begin{array}{ll}
(a) \ [\alpha](\varphi \wedge \psi) \leftrightarrow ([\alpha]\varphi \wedge [\alpha]\psi) & (c) \ [\alpha\beta]\varphi \leftrightarrow [\alpha][\beta]\varphi \\
(b) \ [\alpha \cup \beta]\varphi \leftrightarrow ([\alpha]\varphi \wedge [\beta]\varphi) & (d) \ [\alpha^+]\varphi \leftrightarrow [\alpha](\varphi \wedge [\alpha^+]\varphi)
\end{array}$$

**Proof.** To prove that  $\varphi \leftrightarrow \psi$  is valid it suffices to show that  $V(\varphi, s) = V(\psi, s)$  for all  $s$  in all models. (a) The proof relies on the fact that  $a \Rightarrow (b \sqcap c) = (a \Rightarrow b) \sqcap (a \Rightarrow c)$  in all FL-algebras. (b) The proof relies on the fact that  $(a \sqcup b) \Rightarrow c = (a \Rightarrow c) \sqcap (b \Rightarrow c)$  in all FL-algebras. (c) The proof relies on the fact that  $a \Rightarrow (b \Rightarrow c) = ab \Rightarrow c$  in all FL-algebras. (Note that composition of relations needs to be defined using monoid multiplication  $\cdot$ , not lattice meet.) (d) The proof relies on the fact that  $R_\alpha st \sqsubseteq R_{\alpha^+} st$ , it also uses simple composition of paths.  $\square$

We will discuss an informal interpretation of a special case of the many-valued framework in the next section. Speaking generally, however, we may adapt the slogan characterizing modal logic as providing languages for talking about relational structures [3, p. viii] and say that many-valued modal logics provide *simple yet expressive languages for talking about many-valued relational structures*. Examples of many-valued relational structures include weighted structures such as weighted graphs etc. Choosing an FL-algebra as the algebra of weights brings the framework closer to substructural logics that include well-known formalisms for reasoning about resources (variants of linear logic) or graded properties and relations (fuzzy logics). Many-valued PDL adds to this the capacity to articulate reasoning about *structured* many-valued relations using the PDL relational operations of choice, composition and iteration. An intriguing connection here is the relation of finitely-valued PDL to weighted automata over finite semirings [9], but a more thorough investigation of this connection is left for another occasion.

### 3 Motivation

This section discusses the informal interpretation of finitely-valued PDL. We give two general interpretations of the framework first and then we zoom in to PDLs over a specific class of FL-algebras. Our overview is cursory; the present paper is focused more on basic technical results than on informal interpretations and applications. A more thorough exploration of the latter is left for another occasion. We only note here that we consider many-valued PDL to be sufficiently mathematically interesting to be studied independently of informal interpretations and applications.

We have mentioned before the slogan that modal logics provide simple yet expressive languages for talking about relational structures [3, p. viii]; by the same token, many-valued modal logics can be seen as providing means of talking about “weighted” relational structures. Two-valued PDL has been applied to at least two kinds of relational structures which have very natural weighted generalizations. We discuss these in turn.

First, take the interpretation of modal logic that relates it to *description logics* [2]. Simply put, formulas of a modal language can be seen as express-



ing “concepts”, i.e. properties of objects, and indices of modal operators as expressing various “roles”, i.e. relations between objects. On this reading, “states” in a Kripke model represent arbitrary objects and “accessibility relations” between them represent relations between these objects. Structured modal indices that come with PDL (i.e. “action expressions” as we call them) can be seen as expressing structured relations between objects; union, composition and transitive closure have been found particularly suitable for expressing various important concepts and roles [1]. Many-valued description logics (see [26] for instance) are a generalization of description logics designed for management of *uncertain and imprecise information*. These logics can express the fact that an object is subsumed under a given concept (e.g. “tall” if the reader will forgive the platitudinous example) only to some degree or that only imprecise information about a relation holding between two objects is available. Finitely-valued PDL as presented here can be seen as a family of many-valued description logics with transitive closure of roles.

Second, the original motivation of PDL was reasoning about the behaviour of computer programs [10]. From a more general perspective, PDL can be seen as a logic formalising reasoning about types of *structured actions*, represented by “action expressions”. On this reading, a Kripke frame consists of states and transitions between states labelled by types of action; for instance  $R_\alpha st$  means that action of type  $\alpha$  can be used to get from state  $s$  to state  $t$ . States can be thought of as physical locations, states of a complex system such as a database or states of a computer during the run of a program; but states can also be thought of as “states of the world” that can be modified by actions of intelligent agents. PDL can be used to formalize reasoning about properties of actions that modify these kinds of states. One important example is correctness, related to the question if a specific kind of action is guaranteed to lead to a specific outcome when performed under specific circumstances. (This more general perspective makes PDL relevant to automated planning, for example.) Many-valued Kripke models can be seen as transition systems where transitions carry *weights*; these can be costs or resources needed to perform a transition using the given action type. Běhouněk [5] suggested a many-valued version of PDL for reasoning about costs of program runs that is close to our framework, but he did not establish completeness or decidability results.

Let us now discuss a special case of the finitely-valued PDL framework giving rise to a natural class of weighted relational structures; we show that formulas of the PDL language are able to express interesting features of these structures. Let  $\mathbf{N}$  be the FL-algebra of Example 2.4, that is,  $\mathbf{N} = (N, \max, \min, +_N, \rightarrow_N)$  where

$$a +_N b = \min(a + b, N - 1) \quad \text{and} \quad a \rightarrow_N b = \max(b - a, 0),$$

where  $N \in \omega$  is non-empty. The set  $N$  is seen as a *weight scale* with 0 representing zero weight (“for free”) and  $N - 1$  representing the maximal weight (considered “infeasible”). The operation  $+_N$ , namely, sum bounded by the maximal weight, represents *weight addition*.  $N$  is given a (distributive) lattice

structure by including *max* as meet and *min* as join; the associated lattice order  $\sqsubseteq$  is defined as usual,  $a \sqsubseteq b$  iff  $\min(a, b) = a$ . Hence,  $a \sqsubseteq b$  (i.e.  $b \leq a$ ) means that weight  $b$  is at most as big as weight  $a$ . The choice of *max* as meet and *min* as join—not the other way around—may seem unintuitive at first, but it yields the result that  $a \sqsubseteq 0$  for all  $a \in N$ . It is important to note in this respect that 0 is the identity element with respect to  $+_N$ . (Hence, choosing the natural ordering on  $N$  as our lattice ordering would mean that each element of the lattice would be above the monoid identity, which is problematic given our definition of validity.) It is clear that  $a +_N b = b +_N a$ . The residual  $\rightarrow_N$  of  $+_N$  is truncated subtraction or monus; the crucial feature of  $\rightarrow_N$  is that  $a \rightarrow_N b = 0$  iff  $a \sqsubseteq b$  (iff  $b \sqsubseteq a$ ). We note that  $\mathbf{N}$  is isomorphic to the  $N$ -element Łukasiewicz lattice  $\mathbf{L}_N$  over  $\{\frac{k}{N-1} \mid k \in N\}$ , but we prefer  $\mathbf{N}$  to  $\mathbf{L}_N$  as a representation of an  $N$ -element weight scale.

$\mathbf{N}$ -frames are weighted relational structures that can be informally interpreted in a number of ways. On the “description reading”, for instance, states  $s \in S$  are objects and  $R_\alpha$  represent structured weighted relations between these objects. On the “transition cost reading”, states can be seen as physical locations or states of a system and  $R_\alpha st \in N$  is the cost of accessing state  $t$  from  $s$  by performing action  $\alpha$  (hence, frames are weighted labelled transition systems). If  $R_\alpha st = N - 1$ , then we say that  $t$  is not in relation  $\alpha$  with  $s$ , or that  $t$  cannot be accessed from  $s$  by performing  $\alpha$ ; if  $R_\alpha st = 0$ , then  $t$  is “clearly” in relation  $\alpha$  with  $s$ , or  $t$  can be accessed from  $s$  by  $\alpha$  for free. Let us now discuss some properties of weighted relational structures that can be expressed by PDL formulas.

Since  $\mathbf{N}$  is  $(N - 1)$ -involutive, i.e.  $(a \Rightarrow (N - 1)) \Rightarrow (N - 1) = a$  for all  $a \in \mathbf{N}$ , we have

$$V(\langle \alpha \rangle \bar{0}, s) = \bigsqcup_{t \in S} (R_\alpha st +_N 0) = \min\{R_\alpha st \mid t \in S\}.$$

In other words,  $V(\langle \alpha \rangle \bar{0}, s)$  is the minimal guaranteed cost of performing  $\alpha$  at  $s$  (on the transition cost reading) or the maximal degree to which  $s$  is  $\alpha$ -related to any object (on the description reading). Let us write simply  $\alpha$  instead of  $\langle \alpha \rangle \bar{0}$  if the context clears up any possible confusion. Note that  $a \Rightarrow b$  is the difference between  $b$  and  $a$  if  $a < b$  and 0 otherwise. The following features of weighted relation structures can be expressed (we use the transition cost reading and the reader is invited to translate to the description reading):

- the minimal cost of performing  $\alpha$  is at most  $m$  (this is true in state  $s$  if  $V(\bar{m} \rightarrow \alpha, s) = 0$ ); the “at least” direction is expressed dually;
- performing  $\alpha$  is at least as costly as performing  $\beta$  (this is true in state  $s$  if  $V(\alpha \rightarrow \beta, s) = 0$ ); the “at most” direction is expressed dually;
- the difference between the minimal guaranteed cost of  $\beta$  and  $\alpha$  is at most  $m$  (this is true in state  $s$  if  $V(\bar{m} \rightarrow (\alpha \rightarrow \beta), s) = 0$ ).

On the transition cost reading, atomic formulas in  $Pr$  can be seen as representing various items that can be obtained at states for a given cost, with

$V(p, s)$  representing the cost of item  $p$  at  $s$  (e.g. time needed to charge the battery at the charger location). Observe that  $V(\langle\alpha\rangle p, s) = \bigsqcup_{t \in S} (R_\alpha s t +_N V(p, t))$  is the minimal cost of getting from  $s$  to a state  $t$  by performing  $\alpha$  and obtaining  $p$  at  $t$ ; we may also say that this is the minimal guaranteed cost of obtaining  $p$  by  $\alpha$ . On the description reading, atomic formulas can be seen as expressing graded, imprecise or vague properties of objects; thus the value of  $\langle\alpha\rangle p$  at  $s$  is the “grade of truth” of the statement that  $s$  is  $\alpha$ -related to an object with property  $p$ . The interesting case obtains where both the relation and the property are graded or vague; think of “Alice was in contact with a person displaying symptoms of COVID-19”. We write  $\varphi^\alpha$  instead of  $\langle\alpha\rangle\varphi$ . The following features of weighted relation structures can be expressed (we use the transition cost reading and the reader is again invited to translate to the description reading):

- the minimal cost of obtaining  $p$  by  $\alpha$  is at most  $m$  (this is true in state  $s$  if  $V(\bar{m} \rightarrow p^\alpha, s) = 0$ ); the “at least” direction is expressed dually;
- obtaining  $p$  by  $\alpha$  is at least as costly as obtaining  $q$  by  $\beta$  (this is true in state  $s$  if  $V(p^\alpha \rightarrow q^\beta, s) = 0$ ); the “at most” direction is expressed dually;
- the difference between the minimal guaranteed cost of obtaining  $q$  by  $\beta$  and obtaining  $p$  by  $\alpha$  is at most  $m$  (this is true in state  $s$  if  $V(\bar{m} \rightarrow (p^\alpha \rightarrow q^\beta), s) = 0$ ).

This cursory overview shows that the PDL language provides means to expressing a variety of features of weighted relational structures and so finitely-valued PDL can be used to formalize reasoning about these features. A more thorough exploration of expressivity and applications is left for another occasion.

## 4 Finite model property and decidability

In this section we prove that  $Th(\mathbf{X})$  is decidable for all finite  $\mathbf{X}$ . We prove this by showing that each such  $Th(\mathbf{X})$  has the bounded finite model property. The result is established using a many-valued generalization of the smallest filtration construction; see [8], where the construction is applied to some many-valued modal logics with  $\Box$  and  $\Diamond$ .<sup>2</sup> Even though the decidability result is not surprising, we consider it to be a “sanity check” for the many-valued dynamic framework. We note that presence of canonical constants is not necessary for the decidability result (in contrast to the completeness result of §5).

**Definition 4.1** The *closure* of a set of formulas  $\Psi$  is the smallest  $\Phi \supseteq \Psi$  such that

- $\Phi$  is closed under subformulas (that is, if  $\varphi \in \Phi$  and  $\psi$  is a subformula of  $\varphi$ , then  $\psi \in \Phi$ );
- $[\alpha \cup \beta]\varphi \in \Phi$  implies  $[\alpha]\varphi \in \Phi$  and  $[\beta]\varphi \in \Phi$ ;

<sup>2</sup> We are grateful to an anonymous reviewer for pointing the reference out.

- $[\alpha\beta]\varphi \in \Phi$  implies  $[\alpha][\beta]\varphi \in \Phi$ ;
- $[\alpha^+]\varphi \in \Phi$  implies  $[\alpha][\alpha^+]\varphi \in \Phi$  and  $[\alpha]\varphi \in \Phi$ .

$\Phi$  is *closed* iff  $\Phi$  is the closure of  $\Phi$ .

**Definition 4.2** For each set of formulas  $\Phi$  and each model  $\mathfrak{M}$ , we define the binary two-valued equivalence relation  $\approx_\Phi$  on states of  $\mathfrak{M}$  by

$$s \approx_\Phi t \iff (\forall \varphi \in \Phi)(V(\varphi, s) = V(\varphi, t)).$$

The equivalence class of  $s$  under  $\approx_\Phi$  will be denoted as  $[s]_\Phi$  or just as  $[s]$  if  $\Phi$  is clear from the context.

**Definition 4.3** Take an  $\mathbf{X}$ -valued model  $\mathfrak{M}$  and a finite closed set  $\Phi$ . The *filtration of  $\mathfrak{M}$  through  $\Phi$*  is the  $\mathbf{X}$ -valued model  $\mathfrak{M}^\Phi = (S^\Phi, R^\Phi, V^\Phi)$  such that

- $S^\Phi = \{[s] \mid s \in S\}$ ;
- $R_{a_m}^\Phi([s], [t]) = \bigsqcup \{R_{a_m}(u, v) \mid s \approx_\Phi u \ \& \ t \approx_\Phi v\}$ ;  $R_\alpha^\Phi$  for  $\alpha \notin Ac$  is defined as in models;
- $V^\Phi(p, [s]) = V(p, s)$  for  $p \in \Phi$ ;  $V^\Phi(p, [s]) = 0^{\mathbf{X}}$  for  $p \notin \Phi$ ;  $V^\Phi(\varphi, [s])$  for  $\varphi \notin Pr$  is defined as in models.

It is clear that if  $\Phi$  is the closure of a finite set  $\Psi$ , then  $\Phi$  is finite. If  $\Phi$  is finite, then so is  $\mathfrak{M}^\Phi$ ; in fact,  $|S^\Phi| \leq |\mathbf{X}|^{|\Phi|}$ . We usually omit reference to  $\Phi$  while discussing accessibility relations on  $S^\Phi$  and we also write  $\approx$  instead of  $\approx_\Phi$ . We will write  $R_m$  instead of  $R_{a_m}$ . In the rest of the section, we fix an  $\mathbf{X}$ -model  $\mathfrak{M}$  and a finite closed set  $\Phi$ .

**Lemma 4.4** For all  $\alpha \in ACT$  and all  $x, y \in S$ ,

- (a)  $R_\alpha xy \sqsubseteq R_\alpha[x][y]$ ;
- (b) For all  $[\alpha]\varphi \in \Phi$ ,  $V([\alpha]\varphi, x) \sqsubseteq R_\alpha[x][y] \Rightarrow V(\varphi, y)$ .

**Proof.** Both claims are established by induction on the complexity of  $\alpha$ . The base case of (a) holds by definition and the rest is established easily using the induction hypothesis. In the case of  $\alpha = \beta^+$ , we define for each  $\pi \in \Pi(S)$  of length  $n$  the sequence  $[\pi] \in \Pi(S^\Phi)$  of length  $n$  by  $[\pi](k) := [\pi(k)]$  for all  $k < n$ ; it is then easy to establish by induction on  $n$  that  $R_\beta x \pi y \sqsubseteq R_\beta[x][\pi][y]$ .

The base case of (b) is follows from the fact that, for all  $x' \in [x]$  and  $y' \in [y]$ ,  $\bigsqcup_{z \in S} (R_m x' z \Rightarrow V(\varphi, z)) \cdot R_m x' y' \sqsubseteq V(\varphi, y')$  using the definition of  $\approx_\Phi$ , closure of  $\Phi$  under subformulas and properties of FL-algebras. The fact itself follows easily from properties of FL-algebras. The induction step uses Lemma 2.14 and is easy; for instance, in the case  $\alpha = \beta^+$  we may use the fact that, for all  $x$  and  $y$ ,  $V([\beta](\varphi \wedge [\beta^+]\varphi), x) \sqsubseteq R_\beta[x][y] \Rightarrow V([\beta^+]\varphi, y)$  and hence, for all  $s, t$  and  $\pi \in \Pi(S)$ ,  $V([\beta^+]\varphi, s) \sqsubseteq R_\beta[s][\pi][t] \Rightarrow V(\varphi, t)$  as required.  $\square$

**Lemma 4.5** For all models  $\mathfrak{M}$ , all  $\varphi \in \Phi$  and  $s \in \mathfrak{M}$ ,  $V(\varphi, s) = V^\Phi(\varphi, [s])$ .

**Proof.** The proof is by induction on the complexity of  $\varphi$ . The base case  $\varphi \in Pr$  holds by definition, the cases for constants and propositional connectives are trivial and the case  $\varphi = [\alpha]\psi$  is established using Lemma 4.4.  $\square$

**Theorem 4.6** *Th(X) is decidable for each finite X.*

**Proof.** Lemma 4.5 implies  $\varphi \in Th(\mathbf{X})$  iff  $\varphi$  is valid in all frames where  $|S| \leq |\mathbf{X}|^{|\Phi|}$  where  $\Phi$  is the closure of  $\{\varphi\}$ . Now  $m := |\mathbf{X}| = m$ ,  $n := m^{|\Phi|}$  and let  $n$ -frames be the frames with  $|S| \leq n$ . There are at most

$$n \times m^{n^2}$$

$n$ -frames. On each  $n$ -frame, there are  $n \times m^\omega$  models, but there are at most  $n \times |\Phi| \times m$  possible ways to evaluate elements of  $|\Phi|$  on an  $n$ -frame. Hence, there are at most

$$m^{n^2+1} \times n^2 \times |\Phi|$$

models to check. It is not hard to show that there is an algorithm checking validity of formulas in finite models.  $\square$

## 5 Completeness

Bou et al. [4] establish a general weak completeness result for modal logics based on finite commutative integral FL-algebras with canonical constants where 0 is the bottom element. In this section we build on their work to show how a Hilbert-style axiomatic presentation of any finite commutative integral FL-algebra  $\mathbf{X}$  with canonical constants can be extended to a sound and weakly complete axiomatization of PDL based on  $\mathbf{X}$ . The restriction to commutative FL-algebras seems to be necessary for our style of argument to go through and we discuss this at appropriate places in more detail; the restriction to integral FL-algebras is convenient. We leave generalizations of our result as an open problem.

Fix a finite commutative integral FL-algebra  $\mathbf{X}$  with canonical constants denoting elements of  $\mathbf{X}$ , together with a Hilbert-style axiomatic presentation  $\text{Log}(\mathbf{X})$  in the language  $\mathcal{L}_{\mathbf{X}}$  that is *strongly complete with respect to X*. That is, we assume that  $\varphi \in \mathcal{L}_{\mathbf{X}}$  is derivable from  $\Gamma \subseteq \mathcal{L}_{\mathbf{X}}$  in  $\text{Log}(\mathbf{X})$ , in symbols  $\Gamma \vdash_{\text{Log}(\mathbf{X})} \varphi$ , iff each non-modal homomorphism  $u : \mathcal{L}_{\mathbf{X}} \rightarrow \mathbf{X}$  such that  $1 \sqsubseteq \prod u[\Gamma]$  satisfies  $1 \sqsubseteq u(\varphi)$  (values  $u([\alpha]\psi)$  of modal formulas under  $u$  are arbitrary, so  $u$  “treats” modal formulas as propositional atoms).<sup>3</sup> For the details on how  $\text{Log}(\mathbf{X})$  looks like, see [4]. Since  $\mathbf{X}$  is finite,  $\vdash_{\text{Log}(\mathbf{X})}$  is *finitary* in the sense that if  $\Gamma \vdash_{\text{Log}(\mathbf{X})} \varphi$ , then there is a finite  $\Delta \subseteq \Gamma$  such that  $\Delta \vdash_{\text{Log}(\mathbf{X})} \varphi$ . We note that  $\vdash_{\text{Log}(\mathbf{X})}$  is also *monotonic* in the sense that if  $\Gamma \vdash_{\text{Log}(\mathbf{X})} \varphi$  and  $\Gamma \subseteq \Delta$ , then  $\Delta \vdash_{\text{Log}(\mathbf{X})} \varphi$ .

Since  $\mathbf{X}$  is commutative, we have  $a \setminus b = b/a$  and so we use only a single “official” implication operator  $\rightarrow$ ; see [13, p. 95]. Recall that  $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$ ; we define similarly  $a \Leftrightarrow b := (a \Rightarrow b) \sqcap (b \Rightarrow a)$ .

<sup>3</sup> A function  $f : \mathcal{L}_{\mathbf{X}} \rightarrow \mathbf{X}$  is a non-modal homomorphism iff  $f(\bar{c}) = c$  and  $f$  commutes with the propositional connectives  $\oplus$  of  $\mathcal{L}_{\mathbf{X}}$  and the corresponding operations  $\oplus^{\mathbf{X}}$  on  $\mathbf{X}$ ; we assume that  $\wedge^{\mathbf{X}}$  is  $\sqcap$  and  $\vee^{\mathbf{X}}$  is  $\sqcup$ .

**Definition 5.1**  $\text{PDL}(\mathbf{X})$  is the Hilbert-style axiom system extending  $\text{Log}(\mathbf{X})$  with the following axioms and rules (for all formulas  $\varphi, \psi$ , all action expressions  $\alpha, \beta \in \text{ACT}$  and all canonical constants  $\bar{c}$ ):

$$\begin{array}{ll}
(\text{A-1}) & [\alpha]\bar{1} \\
(\text{A-reg}) & [\alpha]\varphi \wedge [\alpha]\psi \rightarrow [\alpha](\varphi \wedge \psi) \\
(\text{A-}\bar{c}) & [\alpha](\bar{c} \rightarrow \varphi) \leftrightarrow (\bar{c} \rightarrow [\alpha]\varphi) \\
(\text{R-mon}) & \frac{\varphi \rightarrow \psi}{[\alpha]\varphi \rightarrow [\alpha]\psi} \\
(\text{A-}\cup) & [\alpha \cup \beta]\varphi \leftrightarrow ([\alpha]\varphi \wedge [\beta]\varphi) \\
(\text{A-};) & [\alpha\beta]\varphi \leftrightarrow [\alpha][\beta]\varphi \\
(\text{A-}+) & [\alpha^+]\varphi \leftrightarrow [\alpha](\varphi \wedge [\alpha^+]\varphi) \\
(\text{R-}+) & \frac{\varphi \rightarrow [\alpha]\varphi}{\varphi \rightarrow [\alpha^+]\varphi}
\end{array}$$

The notions of proof, derivability, theorem and a formula derivable from a set of formulas are defined as usual (see [4]).  $\text{Thm}(\text{PDL}(\mathbf{X}))$  is the set of theorems of  $\text{PDL}(\mathbf{X})$ .

Since  $\mathbf{X}$  is fixed, we write  $\mathbf{L}$  instead of  $\text{Log}(\mathbf{X})$ ,  $\text{PDL}$  instead of  $\text{PDL}(\mathbf{X})$ ,  $\text{Thm}$  instead of  $\text{Thm}(\text{PDL}(\mathbf{X}))$  and  $\mathcal{L}$  instead of  $\mathcal{L}_{\mathbf{X}}$  for the rest of this section.

**Theorem 5.2** *If  $\varphi$  is a theorem of  $\text{PDL}$ , then  $\varphi$  is valid in the class of all  $\mathbf{X}$ -frames.*

**Proof.** The axioms and the rule in the left column are taken from [4]. Validity of the axioms in the right column in all FL-algebras was established in Lemma 2.14. To show that the rule (R-+) preserves validity in models, assume that  $V(\varphi, s) \sqsubseteq V([\alpha]\varphi, s)$  for all  $s$  in an arbitrary model. Take some  $t$  and assume that  $a \sqsubseteq V(\varphi, t)$ ; we prove that  $a \sqsubseteq R_{\alpha^+}tu \Rightarrow V(\varphi, u)$  for all  $u$ . The claim to be proved is equivalent to  $(\forall \pi \in \Pi(S))(a \sqsubseteq R_{\alpha^+}\pi u \Rightarrow V(\varphi, u))$ . This claim is easily established by induction on the length of  $\pi$ .  $\square$

We note that, without the assumption of commutativity, versions of (A- $\bar{c}$ ) are not sound; the axiom is used in the proof of Lemma 5.6 which is in turn applied in most of our arguments below.

From now on, let  $S$  be the set of non-modal homomorphisms  $s : \mathcal{L} \rightarrow \mathbf{X}$  such that  $s[\text{Thm}] = \{1\}$  and let  $\Phi$  be a fixed finite closed set.

**Definition 5.3** The  $\Phi$ -equivalence relation on  $S$  is an  $\mathbf{X}$ -valued binary relation  $\sim_{\Phi}$  on  $S$  defined by

$$s \sim_{\Phi} t \quad := \quad \prod_{\varphi \in \Phi} (s(\varphi) \leftrightarrow t(\varphi)).$$

If  $\Phi$  is clear from the context, we will write  $s \sim t$  or just  $st$  instead of  $s \sim_{\Phi} t$ .

**Lemma 5.4** *The relation  $\sim_{\Phi}$  is an  $\mathbf{X}$ -valued equivalence relation, that is, (a)  $1 \sqsubseteq s \sim s$ , (b)  $s \sim t = t \sim s$  and (c)  $(s \sim t)(t \sim u) \sqsubseteq s \sim u$ , for all  $s, t, u \in S$ .*

**Proof.** Claims (a) and (b) are clear; claim (c) follows from Lemma 2.6.  $\square$

Completeness proofs for two-valued PDL typically use a filtration-like construction of the canonical model, where states are (or boil down to) equivalence classes of states taken from some other structure. A natural approach in our case would be to take “equivalence classes” of non-modal homomorphisms under  $\sim$ , where  $s \sim t$  expresses “how much equivalent”  $s$  and  $t$  are with respect

to  $\Phi$ . However, in our case a simpler approach is available. We take  $S$  itself as the set of states of the canonical model and we refer to  $\Phi$  only in the definition of the canonical  $R_\alpha$ , which is a generalization of the definition of accessibility relations in the greatest filtration of a Kripke model.

**Definition 5.5** The *canonical model modulo*  $\Phi$  is  $\mathfrak{M} = (S, R, V)$  where

- $S$  is the set of non-modal homomorphisms  $s : \mathcal{L} \rightarrow \mathbf{X}$  such that  $s[\text{Thm}] = \{1\}$ ;
- $R_m st := \prod_{[m]\varphi \in \Phi} (s([m]\varphi) \Rightarrow t(\varphi))$  for all  $a_m \in Ac$  and  $R_\alpha st$  for  $\alpha \notin Ac$  is defined as in models;
- $V(p, s) := s(p)$  and  $V(\varphi, s)$  for  $\varphi \notin Pr$  is defined as in models.

We define for each  $\alpha$  the relation  $R_\alpha^\mathcal{L}$  on  $S$  by  $R_\alpha^\mathcal{L} st := \prod_{\varphi \in \mathcal{L}} (s([\alpha]\varphi) \Rightarrow t(\varphi))$ .

Note that  $R_n^\mathcal{L} st \sqsubseteq R_n st$  for all  $a_n \in Ac$  and all  $s, t$  since  $R_n^\mathcal{L}$  “cares” about more formulas.  $R_\alpha^\mathcal{L}$  is the usual canonical many-valued accessibility relation, see [4], but we cannot use it here because of the presence of the Kleene plus iteration operator in *ACT*, similarly as in the case of two-valued PDL.

The following lemma states some properties of  $R_\alpha^\mathcal{L}$  that will be useful in our proofs; the proof of the lemma can be found in [4] (the logics studied there are mono-modal, but the same approach applies here).

**Lemma 5.6** *The following holds for all  $\alpha \in ACT$  and all  $s \in S$  of the canonical model:*

- (a) For all  $t$ ,  $R_\alpha^\mathcal{L} st = \prod_{\varphi \in \mathcal{L}} \{t(\varphi) \mid 1 \sqsubseteq s([\alpha]\varphi)\}$  ([4], Proposition 4.1.);
- (b) For all  $\varphi \in \mathcal{L}$ ,  $s([\alpha]\varphi) = \prod_{u \in S} \{R_\alpha^\mathcal{L} su \Rightarrow u(\varphi)\}$  ([4], Lemma 4.8.).

**Lemma 5.7** *For all  $[\alpha]\varphi \in \Phi$  and all  $s, t \in S$ ,  $s([\alpha]\varphi) \sqsubseteq R_\alpha st \Rightarrow t(\varphi)$ .*

**Proof.** The claim is proved by induction on the complexity of  $\alpha$ . The base case is established as follows. We know that  $s([n]\varphi) \cdot (s([n]\varphi) \Rightarrow t(\varphi)) \sqsubseteq t(\varphi)$ ; from this  $s([n]\varphi) \cdot R_n st \sqsubseteq t(\varphi)$  follows by the definition of  $R_n$ .

The cases of choice and composition in the induction step are straightforward. The case  $\alpha = \beta^+$  is established by showing that, for all  $\pi \in \Pi(S)$ , all  $s, t$ , and all  $\varphi$  such that  $[\beta^+]\varphi \in \Phi$ ,  $s([\beta^+]\varphi) \sqsubseteq R_\beta s\pi t \Rightarrow t(\varphi)$ . This claim, call it (A), follows from the claims ( $s, t$  and  $[\beta^+]\varphi \in \Phi$  are fixed)

- (B)  $s([\beta]\varphi) \sqsubseteq R_\beta st \Rightarrow t(\varphi)$ ;
- (C) for all  $\sigma \in \Pi(S)$  and all  $u$ ,  $s([\beta^+]\varphi) \sqsubseteq R_\beta s\sigma u \Rightarrow u([\beta^+]\varphi)$ .

The proof of (C) is left to the reader; (B) holds by the induction hypothesis.  $\square$

**Lemma 5.8** *For all  $\alpha$  and  $s, t, u$ ,  $R_\alpha su(ut) \sqsubseteq R_\alpha st$ .*

**Proof.** We argue by induction on the complexity of  $\alpha$ . The base case is established as follows. If  $a \sqsubseteq R_n su(ut)$ , then, by definition,  $a \sqsubseteq \prod_{[n]\varphi \in \Phi} (s([n]\varphi) \Rightarrow u(\varphi))(ut)$ . Hence, for all  $[n]\varphi \in \Phi$ ,  $a \sqsubseteq (s([n]\varphi) \Rightarrow u(\varphi))(u(\varphi) \Rightarrow t(\varphi))$  by the definition of  $u \sim t$  and monotonicity of monoid multiplication (also,  $[n]\varphi \in \Phi$  implies  $\varphi \in \Phi$ ). It follows by the properties of FL-algebras that

$a \sqsubseteq (s([n]\varphi) \Rightarrow t(\varphi))$ . Since  $[n]\varphi \in \Phi$  was arbitrary, we obtain  $a \sqsubseteq R_n st$ . All cases of the induction step are easy.  $\square$

**Definition 5.9** For all  $\alpha$  and  $s$ , we define the following formula:

$$R_\alpha s := \bigvee_{x \in S} \left( \overline{R_\alpha s x} \cdot \bigwedge_{\varphi \in \Phi} (\overline{x(\varphi)} \leftrightarrow \varphi) \right)$$

Note that  $R_\alpha s$  is well defined even though  $S$  is infinite – there are only finitely many possible values of  $R_\alpha s x$  for  $x \in S$ , as  $\mathbf{X}$  is finite. Note also that  $t(R_\alpha s) = \bigsqcup_{x \in S} (R_\alpha s x(xt))$ .

**Lemma 5.10** For all  $s, t$  and  $\alpha$ ,  $t(R_\alpha s) = R_\alpha st$ .

**Proof.** First,  $R_\alpha st \sqsubseteq R_\alpha st(tt)$  by Lemma 5.4(a), and  $R_\alpha st(tt) \sqsubseteq \bigsqcup_{x \in S} (R_\alpha s x(xt)) = t(R_\alpha s)$ . Second,  $R_\alpha s x(xt) \sqsubseteq R_\alpha st$  for all  $x \in S$  by Lemma 5.8. Hence,  $\bigsqcup_{x \in S} R_\alpha s x(xt)$  and so  $t(R_\alpha s) \sqsubseteq R_\alpha st$ .  $\square$

**Lemma 5.11** For all  $s, t \in S$  and all  $\alpha \in ACT$ ,  $R_\alpha^\mathcal{L} st \sqsubseteq R_\alpha st$ .

**Proof.** Induction on the complexity of  $\alpha$ . The base case follows from definition. To establish the induction step, we reason by cases. Note that the induction hypothesis is equivalent to the claim that, for all  $\alpha, \beta$  and  $x$ ,  $1 \sqsubseteq x([\alpha]R_\alpha x)$  and  $1 \sqsubseteq x([\beta]R_\beta x)$  by Lemmas 5.6(b) and 5.10.

If  $a \sqsubseteq R_{\alpha \cup \beta}^\mathcal{L} st$ , then  $a \sqsubseteq \prod_{\varphi \in \mathcal{L}} \{t(\varphi) \mid 1 \sqsubseteq s([\alpha \cup \beta]\varphi)\}$  by Lemma 5.6(a). By the definition of  $S$ , this entails  $a \sqsubseteq \prod \{t(\varphi) \mid 1 \sqsubseteq s([\alpha]\varphi) \wedge s([\beta]\varphi)\}$ . By the induction hypothesis,  $1 \sqsubseteq s([\alpha]R_\alpha s)$  and  $1 \sqsubseteq s([\beta]R_\beta s)$ . Hence,  $1 \sqsubseteq s([\alpha](R_\alpha s \vee R_\beta s))$  and  $1 \sqsubseteq s([\beta](R_\alpha s \vee R_\beta s))$  by the definition of  $S$ . It follows that  $a \sqsubseteq t(R_\alpha s) \sqcup t(R_\beta s)$ . By Lemma 5.10,  $a \sqsubseteq R_\alpha st \sqcup R_\beta st$  and so  $a \sqsubseteq R_{\alpha \cup \beta} st$ .

If  $a \sqsubseteq R_{\alpha\beta}^\mathcal{L} st$ , then  $a \sqsubseteq \prod_{\varphi \in \mathcal{L}} \{t(\varphi) \mid 1 \sqsubseteq s([\alpha\beta]\varphi)\}$  by Lemma 5.6(a) and so  $a \sqsubseteq \prod_{\varphi \in \mathcal{L}} \{t(\varphi) \mid 1 \sqsubseteq s([\alpha][\beta]\varphi)\}$  by the definition of  $S$ . For all  $x$  and  $y$ ,  $R_\alpha^\mathcal{L} s x R_\beta^\mathcal{L} x y \sqsubseteq y(R_{\alpha\beta} s)$  by the induction hypothesis, Lemma 5.10 and the definition of  $R_{\alpha\beta}$ . Hence, for all  $x$ ,  $R_\alpha^\mathcal{L} s x \sqsubseteq x([\beta]R_{\alpha\beta} s)$  by residuation and Lemma 5.6(b); from this it follows that  $1 \sqsubseteq s([\alpha][\beta]R_{\alpha\beta} s)$  by another application of residuation and Lemma 5.6(b). Therefore,  $a \sqsubseteq t(R_{\alpha\beta} s)$  and so  $a \sqsubseteq R_{\alpha\beta} st$  by Lemma 5.10.

Finally, we discuss the case of  $\alpha^+$ . Fix  $s$ ; we write  $F$  instead of  $R_{\alpha^+} s$ . Note that  $R_{\alpha^+}$  is a transitive relation extending  $R_\alpha^\mathcal{L}$ . Hence, for all  $t, u \in S$ ,  $u(F) \cdot R_\alpha^\mathcal{L} ut \sqsubseteq t(F)$  by Lemma 5.10 and the induction hypothesis applied to  $R_\alpha^\mathcal{L}$ ; we obtain from this  $u(F) \sqsubseteq u([\alpha]F)$  for all  $u \in S$  by Lemma 5.6(b). Hence, by definition of  $S$ , we have  $F \rightarrow [\alpha]F \in \text{Thm}$ . Hence, using (R-+), we have  $F \rightarrow [\alpha^+]F \in \text{Thm}$  and, using (R-mon) and (A-+), we obtain  $[\alpha]F \rightarrow [\alpha^+]F \in \text{Thm}$ . By the induction hypothesis we have  $R_\alpha^\mathcal{L} st \sqsubseteq R_\alpha st \sqsubseteq R_{\alpha^+} st$  for all  $t$  and so  $1 \sqsubseteq R_\alpha^\mathcal{L} st \Rightarrow t(F)$  for all  $t$  by Lemma 5.10. This means that  $1 \sqsubseteq s([\alpha]F)$  and so  $1 \sqsubseteq s([\alpha^+]F)$  which means that  $R_{\alpha^+}^\mathcal{L} st \sqsubseteq t(F)$  for all  $t$  by Lemma 5.6(b). Hence,  $R_{\alpha^+}^\mathcal{L} st \sqsubseteq R_{\alpha^+} st$  by Lemma 5.10.  $\square$

**Lemma 5.12** For all  $\varphi \in \Phi$  and  $s \in S$ ,  $s(\varphi) = V(\varphi, s)$ .



**Proof.** Induction on the complexity of  $\varphi$ . The base case holds by definition and the cases for non-modal formulas and canonical constants are straightforward. Finally,  $s([\alpha]\varphi) \sqsubseteq V([\alpha]\varphi, s)$  holds thanks to Lemma 5.7 and  $V([\alpha]\varphi, s) \sqsubseteq s([\alpha]\varphi)$  holds thanks to Lemma 5.6(b) and Lemma 5.11.  $\square$

**Theorem 5.13** *For all finite commutative integral  $\mathbf{X}$  with canonical constants,  $\varphi$  is valid in all  $\mathbf{X}$ -frames iff  $\varphi$  is a theorem of  $\text{PDL}(\mathbf{X})$ .*

**Proof.** Soundness is established by Theorem 5.2. Completeness is established as usual. If  $\varphi$  is not in  $\text{Thm}$ , then  $\text{Thm} \not\vdash_{\mathbf{L}} \varphi$  since  $\text{Thm}$  is obviously closed under  $\vdash_{\mathbf{L}}$ . By strong completeness of  $\mathbf{L}$ , there is a non-modal homomorphism from  $\mathcal{L}$  to  $\mathbf{X}$  such that  $s[\text{Thm}] = \{1\}$  and  $s(\varphi) \neq 1$ . Let  $\Phi$  be the closure of  $\{\varphi\}$ ;  $\varphi$  is not valid in the canonical model modulo  $\Phi$  by Lemma 5.12.  $\square$

### 6 On Kleene star and test

Our syntactic presentation of propositional dynamic logic differs from the standard presentation in two important respects, namely, (i) our action operators do not include the *Kleene star*, but rather the Kleene plus operator; (ii) we do not include the *test operator*. Kleene star and test are instrumental in the ability of classical PDL to express standard programming constructs such as while loops and conditionals (test suffices for the latter). In this section we discuss these omissions.

Concerning the Kleene star, Proposition 2.13 suggests that, working with frames based on finite integral FL-algebras, we can define, for all  $\alpha \in \text{ACT}$  and  $\varphi \in \text{Fm}(\mathcal{L}_{\mathbf{X}})$ ,

$$[\alpha^*]\varphi := [\alpha^+]\varphi \wedge \varphi$$

as a semantically equivalent surrogate for formulas with the Kleene star. For instance,  $[(a \cup b)^*; a^*]p$  is short for  $[(a \cup b)^+][a^+]p \wedge p \wedge ([a^+]p \wedge p)$ . However, it is clear that not all action expressions in *STA* can be expressed by action expressions in *ACT*. Therefore, for example,  $[(a^*; b)^*]p$  is not a well-formed formula since  $a^* \notin \text{ACT}$ .

The technical problem that precluded us from working with Kleene star as a primitive operator is related to Lemma 5.8. Take the reflexive transitive closure  $R_{\alpha}^*$  of  $R_{\alpha}$ , defined as in Proposition 2.8. The issue is that Lemma 5.8 fails if Kleene star is a primitive operator and we define  $R_{\alpha^*} := R_{\alpha}^*$ . In particular, if  $s = u \neq t$ , then  $R_{\alpha}^*su(ut) \sqsubseteq R_{\alpha}^*st$  boils down to  $s \sim t \sqsubseteq R_{\alpha^+}st$ , which does not hold in all canonical models. (Take the canonical  $\mathbf{2}$ -model modulo the closure  $\Phi$  of  $\Psi = \{[a]\perp\}$ . As both  $\Psi \cup \{p_0\}$  and  $\Psi \cup \{p_1\}$  are consistent, there are two distinct  $s, t$  such that  $s \sim_{\Phi} t$  equals 1, but  $R_{a^+}st$  equals 0.)

Concerning test, a natural semantic interpretation of  $\varphi?$ , endorsed also in [18,19], is

$$R_{\varphi?}(s, t) = \begin{cases} V(\varphi, s) & \text{if } s = t \\ \perp^{\mathbf{X}} & \text{otherwise.} \end{cases}$$

However, Lemma 5.8 turns out to be problematic for such a relation as well. (Take the model from the previous paragraph and let  $\varphi = [a]\perp$ ; clearly

$R_{\varphi?ss}(st)$  equals 1, but  $R_{\varphi?st}$  equals 0.)

It is clear that a more substantial modification of our completeness argument is needed to accommodate logics with Kleene star and test. This is an interesting problem we leave open here.

## 7 Conclusion

We have studied a general framework for many-valued versions of Propositional Dynamic Logic where both formulas in states and accessibility relations between states of a Kripke model are evaluated in a finite FL-algebra. We established a general decidability result and we provided a general completeness argument for PDLs based on commutative integral FL-algebras with canonical constants. We build on previous work on many-valued modal logic and our techniques are generalizations of the arguments used in the two-valued case; however, to the best of our knowledge, the technical results presented here are the first decidability and completeness results on PDL with many-valued accessibility relations. As our discussion of the informal interpretations of the framework suggests, many-valued PDL has links to existing research in description logics and potential applications in reasoning about weighted labelled transition systems.

Our paper also suggests a number of topics for future research. We would like to mention especially the addition of test and further work on the standard version of PDL with primitive Kleene star in the many-valued setting. Another topic are generalizations of our results beyond finite (commutative integral) FL-algebras with canonical constants; in many cases the work here would require modifications of existing techniques used in completeness arguments for many-valued modal logics without “structured” modal operators. Finally, informal interpretations and applications of our framework need to be explored in more detail.

## References

- [1] Baader, F., *Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles*, in: *IJCAI'91*, 1991, pp. 446–451.
- [2] Baader, F., D. Calvanese, D. L. McGuinness, D. Nardi and P. F. Patel-Schneider, editors, “The Description Logic Handbook: Theory, Implementation, and Applications,” Cambridge University Press, 2007, 2nd edition edition.
- [3] Blackburn, P., M. de Rijke and Y. Venema, “Modal Logic,” Cambridge University Press, 2001.
- [4] Bou, F., F. Esteva, L. Godo and R. O. Rodríguez, *On the minimum many-valued modal logic over a finite residuated lattice.*, *Journal of Logic and Computation* **21** (2011), pp. 739–790.
- [5] Běhounek, L., *Modeling costs of program runs in fuzzified propositional dynamic logic*, in: F. Haki, editor, *Doktorandské dny '08* (2008), pp. 6 – 14.
- [6] Caicedo, X. and R. O. Rodríguez, *Standard Gödel modal logics*, *Studia Logica* **94** (2010), pp. 189–214.
- [7] Caicedo, X. and R. O. Rodríguez, *Bi-modal Gödel logic over [0,1]-valued Kripke frames*, *Journal of Logic and Computation* **25** (2015), pp. 37–55.

- [8] Conradie, W., W. Morton and C. Robinson, *Filtrations for many-valued modal logic with applications* (2017), presentation at TACL 2017, Prague.
- [9] Droste, M., W. Kuich and H. Vogler, editors, “Handbook of Weighted Automata,” Springer, 2009.
- [10] Fischer, M. J. and R. E. Ladner, *Propositional dynamic logic of regular programs*, Journal of Computer and System Sciences **18** (1979), pp. 194–211.
- [11] Fitting, M., *Many-valued modal logics*, Fundamenta Informaticae **15** (1991), pp. 235–254.
- [12] Fitting, M., *Many-valued modal logics II*, Fundamenta Informaticae **17** (1992), pp. 55–73.
- [13] Galatos, N., P. Jipsen, T. Kowalski and H. Ono, “Residuated Lattices: An Algebraic Glimpse at Substructural Logics,” Elsevier, 2007.
- [14] Hájek, P., “Metamathematics of Fuzzy Logic,” Kluwer, 1998.
- [15] Hansoul, G. and B. Teheux, *Extending Łukasiewicz logics with a modality: Algebraic approach to relational semantics*, Studia Logica **101** (2013), pp. 505–545.
- [16] Harel, D., “First-Order Dynamic Logic,” Lecture Notes in Computer Science **68**, Springer, 1979.
- [17] Harel, D., D. Kozen and J. Tiuryn, “Dynamic Logic,” MIT Press, 2000.
- [18] Hughes, J., A. Esterline and B. Kimiaghalam, *Means-end relations and a measure of efficacy*, Journal of Logic, Language and Information **15** (2006), pp. 83–108.
- [19] Liau, C.-J., *Many-valued dynamic logic for qualitative decision theory*, in: N. Zhong, A. Skowron and S. Ohsuga, editors, *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing* (1999), pp. 294–303.
- [20] Madeira, A., R. Neves and M. A. Martins, *An exercise on the generation of many-valued dynamic logics*, Journal of Logical and Algebraic Methods in Programming **85** (2016), pp. 1011–1037.
- [21] Madeira, A., R. Neves, M. A. Martins and L. S. Barbosa, *A dynamic logic for every season*, in: C. Braga and N. Martí-Oliet, editors, *Formal Methods: Foundations and Applications* (2015), pp. 130–145.
- [22] Odintsov, S. and H. Wansing, *Modal logics with Belnapian truth values*, Journal of Applied Non-Classical Logics **20** (2010), pp. 279–301.
- [23] Ostermann, P., *Many-valued modal propositional calculi*, Mathematical Logic Quarterly **34** (1988), pp. 343–354.
- [24] Sedlár, I., *Propositional dynamic logic with Belnapian truth values*, in: *Advances in Modal Logic. Vol. 11* (2016).
- [25] Segerberg, K., *Some modal logics based on a three-valued logic*, Theoria **33** (1967), pp. 53–71.
- [26] Straccia, U., *Description logics over lattices*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **14** (2006), pp. 1–16.
- [27] Teheux, B., *Propositional dynamic logic for searching games with errors*, Journal of Applied Logic **12** (2014), pp. 377–394.
- [28] Vidal, A., F. Esteva and L. Godo, *On modal extensions of product fuzzy logic*, Journal of Logic and Computation **27** (2017), p. 299.