Decidability and Complexity of Some Finitely-valued Dynamic Logics

Igor Sedlár
Czech Academy of Sciences, Institute of Computer Science
sedlar@cs.cas.cz

Abstract

Propositional Dynamic Logic, **PDL**, is a well known modal logic formalizing reasoning about complex actions. We study many-valued generalizations of **PDL** based on relational models where satisfaction of formulas in states and accessibility between states via action execution are both seen as graded notions, evaluated in a finite Łukasiewicz chain. For each $n > 1$, the logic $\mathbf{PDŁ}_n$ is obtained using the n-element Łukasiewicz chain, **PDL** being equivalent to $\mathbf{PDŁ}_2$. These finitely-valued dynamic logics can be applied in formalizing reasoning about actions specified by graded predicates, reasoning about costs of actions, and as a framework for certain graded description logics with transitive closure of roles. Generalizing techniques used in the case of **PDL** we obtain completeness and decidability results for all $\mathbf{PDŁ}_n$. A generalization of Pratt's exponential-time algorithm for checking validity of formulas is given and $EXPTIME$-hardness of each $\mathbf{PDŁ}_n$ validity problem is established by embedding **PDL** into $\mathbf{PDŁ}_n$.

1 Introduction

Propositional Dynamic Logic, **PDL**, is a modal logic originally introduced to formalize reasoning about correctness of imperative programs and equivalence of regular expressions (Fischer and Ladner 1979; Harel, Kozen, and Tiuryn 2000). **PDL** represents programs as binary relations on states and uses specific operations on programs, most notably sequential composition $a; b$ ("do a and then do b", represented as relational composition), choice $a \cup b$ ("do a or do b", union), Kleene star a^* ("do a some finite number of times", reflexive transitive closure) and test $p?$ ("test if p is true", the smallest reflexive relation on states satisfying p). **PDL** subsumes Hoare logic, owing to its ability to express programming constructs such as "**if** p **then** a **else** b" or "**while** p **do** a".

PDL has also been applied in formalizing reasoning about complex actions in general, e.g. in planning (Rosenschein 1981; Spalazzi and Traverso 2000) or reasoning about permissibility of actions (Meyer 1987; Meyer 2000). **PDL** is related to a number of formalisms used in Knowledge Representation; it can be seen, via a translation to predicate logic (Blackburn, de Rijke, and Venema 2001), as the variable-free fragment of the situation calculus (McCarthy 1963; Reiter 2001), and it subsumes the extension of the description logic $\mathcal{ALC}$ with composition, union and transitive closure of roles (Baader 1991). The KARO framework uses **PDL** program operators (van der Hoek, van Linder, and Meyer 1994; Meyer, van der Hoek, and van Linder 1999) which also appear in the action component of the logic $\mathcal{LORA}$ (Wooldridge 2000). Epistemic logic with common knowledge (Fagin et al. 1995) is a fragment of **PDL**.

Models of **PDL** are directed labelled graphs, where edges are labelled by programs and nodes by formulas. Intuitively, the edge (s, t) is labelled by a if executing action/program a in state s may result in state t (on the action reading), if object t is connected to s via the role a (the description logic reading), or if state t is epistemically accessible to agent a (the epistemic reading); the node s is labelled by p if the statement represented by p holds in state s (on the action and epistemic readings) or if the individual s satisfies the description expressed by the concept p (on the description logic reading). Crucially, these labels are *crisp*, that is, either applying to an edge/node or not. This is unrealistic in many contexts, for instance if *graded* concepts and roles or *costs* of actions are considered.[1] Versions of **PDL** lifting the crispness assumption are *many-valued* modal logics with models where the degrees to which labels apply to nodes and edges are represented by an algebra extending the two-element Boolean algebra.

Many-valued dynamic logics have been proposed in AI to formalize reasoning about actions with graded goals (Liau 1999) and reasoning involving probabilistic information about the outcomes of actions (Hughes, Esterline, and Kimiaghalam 2006); they have also been suggested as a formalism for reasoning about costs of program runs (Běhounek 2008), weighted computation (Madeira, Neves, and Martins 2016) and for analysing searching games (Teheux 2014). Only a handful of results on axiomatization and decidability of many-valued dynamic logics have been established and, to the best of our knowledge, no results on computational complexity of these logics are available so far. Such results are the main contribution of the present paper.

[1] Graded concepts are "a matter of more-or-less" (Cintula, Noguera, and Smith 2017); e.g. consider the role :IsEasilyAccessibleFrom or the concept GoodRestaurant (restaurants are good to a degree, some being better than others; physical locations are easily accessible from one another to a degree, some being more easy to get to than others). Costs of actions give rise to specific kinds of graded relations between actions and states, such as "it does not take much time to get from s to t by doing a".

We study many-valued dynamic logics based on models where both edge and node labellings are evaluated in a finite linearly ordered algebra called a *finite Łukasiewicz chain*. Logics arising from these models are $\mathbf{PDŁ}_n$, for $n > 1$, depending on the cardinality of the given Łukasiewicz chain, **PDL** being $\mathbf{PDŁ}_2$. We prove for each n that the set of formulas valid in $\mathbf{PDŁ}_n$ is decidable (Theorem 5.12) and that the validity problem is $EXPTIME$-complete (Theorems 6.3 and 6.6); decidability is established by providing a recursive axiomatization (Theorem 5.11) and proving the finite model property using filtration (Theorem 5.10). This extends the related literature as follows. Teheux (2014) studies many-valued dynamic logics based on finite Łukasiewicz chains, proving decidability and establishing a recursive axiomatization, but in his models only evaluation of formulas in states is many-valued and accessibility is crisp. The present author (Sedlár 2020) studies many-valued dynamic logics based on finite Łukasiewicz chains where both evaluation of formulas and accessibility are many-valued, establishing decidability and recursive axiomatization, but only for logics with a limited set of program operators, namely, without test and with a transitive closure operator replacing the more usual reflexive transitive closure operator (Kleene star); complexity results are not established in that paper.[2]

The paper is structured as follows. Section 2 discusses graded relations and outlines basic facts about finite Łukasiewicz chains, $\boldsymbol{Ł}_n$ for $n > 1$. Section 3 introduces many-valued dynamic logics based on finite Łukasiewicz chains, $\mathbf{PDŁ}_n$ for $n > 1$. Section 4 discusses the informal interpretation of $\mathbf{PDŁ}_n$ in more detail, hinting at some possible applications by means of a concrete example formalized using $\mathbf{PDŁ}_n$. Soundness and weak completeness of recursive axiomatizations of $\mathbf{PDŁ}_n$ and decidability of these logics are established in Section 5; the results are obtained using a generalization of the standard filtration-based argument (Kozen and Parikh 1981; Harel, Kozen, and Tiuryn 2000) used in the case of crisp **PDL**. Section 6 contains results on complexity of the validity problem of each $\mathbf{PDŁ}_n$—Pratt's exponential-time algorithm for crisp **PDL** is generalized to suit arbitrary $\mathbf{PDŁ}_n$ and $EXPTIME$-hardness is established by constructing embeddings of **PDL** to $\mathbf{PDŁ}_n$.

[2]We should also mention the paper (Di Nola, Grigolia, and Vitale 2020), sketching an axiomatization result for many-valued dynamic logics based on Łukasiewicz chains using models with many-valued evaluation and crisp accessibility (obtained independently of Teheux, apparently). (Zhang et al. 2012) combine crisp **PDL** with modal epistemic (BDI) logic where epistemic formulas take truth degrees in the real unit interval $[0, 1]$ and sketch a completeness argument. (Liau 1999; Hughes, Esterline, and Kimiaghalam 2006; Běhounek 2008; Madeira, Neves, and Martins 2016) do not establish axiomatization or decidability/complexity results. The $\mathbf{PDŁ}_n$ logics are fragments of the Łukasiewicz μ-calculus (Mio and Simpson 2017), but our main results on the former do not follow from existing results on the latter. $\mathbf{PDŁ}_n$ resemble logics of graded belief-based programs (Laverny and Lang 2005a; Laverny and Lang 2005b), but the latter use "crisp" modal operators indexed by grades, not many-valued interpretations as we do; see also (van der Hoek and Meyer 1992) for a similar modelling of "graded" modalities, differing from our approach.

2 Truth Degrees and Finite Łukasiewicz Chains

2.1 Modelling Truth Degrees

Crisp n-ary relations may be identified with functions from n-tuples of individuals to the two-element set of truth values $\{0, 1\}$, 0 standing for "false" and 1 "for true". It is natural to model *graded* n-ary relations (and, as a special case, *vague* relations) as functions from n-tuples of individuals to some (linearly ordered) set of *truth degrees*. The usual approach is to take an algebra based on the real unit interval $[0, 1]$; see (Cintula, Noguera, and Smith 2017; Hájek 1998; Smith 2008). For instance, $\mathsf{Tall(john)}$ may evaluate to 0.9 if John is 180 cm tall and $\mathsf{Tall(jack)}$ may evaluate to 0.3 if Jack is 170 cm tall.[3] If a truth degree of a statement is $x \in [0, 1]$, then $1 - x$ may be seen as the *cost* associated with the statement. For instance, the cost of $\mathsf{Tall(john)}$ in the previous example is relatively low, 0.1, meaning that we do not have to "stretch" the meaning of Tall too much when ascribing it to John. To take another example, the cost of $\mathsf{CheapFlight(prague,london)}$, a statement asserting the possibility to fly cheaply from Prague to London, may be directly associated with the financial cost of the relevant flights.

A crucial role in any algebra of truth degrees is played by a binary operator representing *merging* of truth degrees; there are several such operations on the market, but here we will focus on the one called the *Łukasiewicz t-norm*: for any $x, y \in [0, 1]$, $x \odot_{Ł} y = \max(0, x + y - 1)$. The Łukasiewicz t-norm represents the idea that the truth degree of "p and q", in one sense of "and", is computed by considering the *combined costs* of p and q. For instance, if "and" is interpreted as the Łukasiewicz t-norm, then the truth degree of "$\mathsf{Tall(john)}$ and $\mathsf{Tall(jack)}$" is 0.2, that is, $1-(0.1+0.7) = (0.9+0.3)-1$.

We emphasize that truth degrees are not to be confused with probabilities, as discussed in (Hájek, Godo, and Esteva 1995) and (Hughes, Esterline, and Kimiaghalam 2006) for instance, the main reason being that truth degrees are truth-functional, i.e. the truth degree of a compound statement is a function of the truth degrees of its components.[4]

In many settings it is impractical to consider the full interval $[0, 1]$ and then one usually takes some finite subalgebra of the $[0, 1]$-based algebra of truth degrees. In the context of $[0, 1]$ with the Łukasiewicz t-norm, this approach yields algebras known as finite Łukasiewicz chains.

2.2 Finite Łukasiewicz Chains

Our discussion of finite Łukasiewicz chains is rather dense, given the space limitations; we invite the reader to discuss

[3]One objection against this modelling of graded relations is that it introduces *artificial precision* as, presumably, there is not one specific function from n-tuples to $[0, 1]$ for each graded relation; see (Smith 2008) for a "pluralistic" elaboration of the modelling in response to the artificial precision objection. Here we ignore the objection for the sake of simplicity, but a combination of our framework with Smith's approach is a topic for future research.

[4]Hence, our dynamic logics differ in nature from Kozen's probabilistic dynamic logic (Kozen 1985) and from probability-based formalisms for reasoning about action under uncertainty, such as (Bacchus, Halpern, and Levesque 1999) for instance.

(Hájek 1998) or (Grigolia 1977).

Definition 2.1. A *finite Łukasiewicz chain* is an algebra $\boldsymbol{Ł}_n = \langle Ł_n, \oplus, \odot, \sim, 1, 0\rangle$ of type $\langle 2,2,1,0,0\rangle$ where $n > 1$ and $Ł_n = \left\{ \frac{k}{n-1} \,\middle|\, 0 \leq k \leq n-1 \right\}$, $x \oplus y = \min(1, x+y)$, $x \odot y = \max(0, x+y-1)$ and $\sim x = 1-x$. For all $\boldsymbol{Ł}_n$ we define $x \to^{\boldsymbol{Ł}_n} y := \sim(x \odot \sim y)$ (it follows that $x \to^{\boldsymbol{Ł}_n} y = \min(1, 1-x+y)$), $x \wedge^{\boldsymbol{Ł}_n} y := x \odot (x \to^{\boldsymbol{Ł}_n} y)$ (= $\min(x,y)$), $x \vee^{\boldsymbol{Ł}_n} y := (x \to^{\boldsymbol{Ł}_n} y) \to^{\boldsymbol{Ł}_n} y$ (= $\max(x,y)$), and $x \sqsubseteq y$ iff $x \vee^{\boldsymbol{Ł}_n} y = y$.

Example 2.2. For instance, $Ł_{11} = \{0, 0.1, \ldots, 0.9, 1\}$; here $0.8 \odot 0.6 = 0.4$, $0.9 \odot 0.9 = 0.8$, and $0.9 \to^{\boldsymbol{Ł}_n} 0.6 = 0.7$.

Hence, $\oplus$ is (truncated) addition and $\odot$ is the Łukasiewicz t-norm. In general, $\sim(x \odot y) = \sim x \oplus \sim y$ (the cost of $x \odot y$ is the truncated sum of the costs of x and y), $x \to^{\boldsymbol{Ł}_n} y = \sim x \oplus y = 1$ if $x \sqsubseteq y$ and $= 1 - x + y$ otherwise. The Łukasiewicz implication operator $\to^{\boldsymbol{Ł}_n}$ is also known as the *residual* of the Łukasiewicz t-norm operation since we have the following residuation law: $x \odot y \sqsubseteq z$ iff $x \sqsubseteq y \to^{\boldsymbol{Ł}_n} z$. We define $x \leftrightarrow^{\boldsymbol{Ł}_n} y := (x \to^{\boldsymbol{Ł}_n} y) \wedge^{\boldsymbol{Ł}_n} (y \to^{\boldsymbol{Ł}_n} x)$; $x^0 := 1$; $x^{n+1} := x^n \odot x$; $0x := 0$ and $(n+1)x := nx \oplus x$.

Lemma 2.3. *In each* $\boldsymbol{Ł}_n$*,*

1. $x^n = \begin{cases} 1 & \text{if } x = 1 \\ 0 & \text{if } x \neq 1 \end{cases}$ *2.* $nx = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases}$

We will often rely on following properties of Łukasiewicz chains without explicit reference:

Lemma 2.4. *The following holds in each Łukasiewicz chain:*

1. $x \to^{\boldsymbol{Ł}_n} y = 1$ *iff* $x \sqsubseteq y$
2. $x \to^{\boldsymbol{Ł}_n} (y \to^{\boldsymbol{Ł}_n} z) \sqsubseteq y \to^{\boldsymbol{Ł}_n} (x \to^{\boldsymbol{Ł}_n} z)$
3. $x \to^{\boldsymbol{Ł}_n} (y \to^{\boldsymbol{Ł}_n} z) = (x \odot y) \to^{\boldsymbol{Ł}_n} z$
4. $\sim\sim x = x$
5. $(x \to^{\boldsymbol{Ł}_n} y) \wedge^{\boldsymbol{Ł}_n} (x \to^{\boldsymbol{Ł}_n} z) = x \to^{\boldsymbol{Ł}_n} (y \wedge^{\boldsymbol{Ł}_n} z)$
6. $(x \to^{\boldsymbol{Ł}_n} z) \wedge^{\boldsymbol{Ł}_n} (y \to^{\boldsymbol{Ł}_n} z) = (x \vee^{\boldsymbol{Ł}_n} y) \to^{\boldsymbol{Ł}_n} z$
7. $x \odot (y \vee^{\boldsymbol{Ł}_n} z) = (x \odot y) \vee^{\boldsymbol{Ł}_n} (x \odot z)$

Let P be a countable set of propositional variables $p_0, p_1, \ldots$. The propositional language $\mathcal{P}_n$ (which we identify with the set of its formulas) is defined by:

$$\varphi \;:=\; p \mid \bar{c} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi$$

where $p \in P$ and $c \in Ł_n$. We define $\neg\varphi := \varphi \to \bar{0}$, $\varphi \leftrightarrow \psi := (\varphi \to \psi) \wedge (\psi \to \varphi)$, $\varphi \& \psi := \neg(\varphi \to \neg\psi)$, and $\varphi \oplus \psi := \neg\varphi \to \psi$. Let $\varphi^0 := \bar{1}$ and $\varphi^{n+1} := \varphi^n \& \varphi$; let $0\varphi := \bar{0}$ and $(n+1)\varphi := n\varphi \oplus \varphi$.

A $\boldsymbol{Ł}_n$*-valuation* is any $v : \mathcal{P}_n \to \boldsymbol{Ł}_n$ such that (i) $v(\bar{c}) = c$ and (ii) $v(\varphi \star \psi) = v(\varphi) \star^{\boldsymbol{Ł}_n} v(\psi)$ for $\star \in \{\wedge, \vee, \to\}$. A function from any $\mathcal{L} \supseteq \mathcal{P}_n$ to $\boldsymbol{Ł}_n$ satisfying (i, ii) will be called an $\boldsymbol{Ł}_n$*-homomorphism* from $\mathcal{L}$ to $\boldsymbol{Ł}_n$.

A formula $\varphi \in \mathcal{P}_n$ is a $\boldsymbol{Ł}_n$*-consequence* of a set of formulas $\Gamma \subseteq \mathcal{P}_n$ iff, for each $\boldsymbol{Ł}_n$-valuation v, if $v(\psi) = 1$ for all $\psi \in \Gamma$, then $v(\varphi) = 1$.

Definition 2.5. For each $n \geq 2$, the Hilbert-style proof system $Ł_n$ is defined as follows. The axiom schemata are:

($Ł_n$1) $\varphi \to (\psi \to \varphi)$
($Ł_n$2) $(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$
($Ł_n$3) $(\neg\varphi \to \neg\psi) \to (\psi \to \varphi)$
($Ł_n$4) $((\varphi \to \psi) \to \psi) \to ((\psi \to \varphi) \to \varphi)$
($Ł_n$5) $\overline{c \star^{\boldsymbol{Ł}_n} d} \leftrightarrow (\bar{c} \star \bar{d})$, for $\star \in \{\wedge, \vee, \to\}$

The sole rule of inference is Modus Ponens. The proof system in fact coincides with the axiomatization of Rational Pavelka Logic (Hájek 1998), with the exception that our set of constants $\{\bar{c} \mid c \in Ł_n\}$ is finite.

We say that φ is derivable from Γ in $Ł_n$ iff there is a finite sequence of formulas $\psi_0, \ldots, \psi_{m-1}$ such that $\psi_{m-1} = \varphi$ and each ψ_i for $i < m$ is either an instance of an axiom schema, an element of Γ or is derived from some previous formulas by Modus Ponens. (Notation: $\Gamma \vdash_{Ł_n} \varphi$.)

Theorem 2.6. *φ is a $\boldsymbol{Ł}_n$-consequence of Γ iff $\Gamma \vdash_{Ł_n} \varphi$.*

Proof. The proof mimics the completeness proof for Rational Pavelka Logic in (Hájek 1998), we omit the details. (The key step in the proof is a Truth Lemma saying that, if Γ is a complete and consistent theory—that is, $\Gamma \nvdash_{Ł_n} \bar{0}$ and if $\Gamma \nvdash_{Ł_n} \varphi \to \psi$, then $\Gamma \vdash_{Ł_n} \psi \to \varphi$—then the function $v_\Gamma : \varphi \mapsto \max\{c \mid \Gamma \vdash_{Ł_n} \bar{c} \to \varphi\}$ is a $\boldsymbol{Ł}_n$-valuation such that $v_\Gamma[\Gamma] = \{1\}$.) □

An $\boldsymbol{Ł}_n$*-valued binary relation on a set S* is a function from $S \times S$ to $\boldsymbol{Ł}_n$. Generalizations of the usual operations on binary relations are defined as expected: (i) Identity relation $\iota(s,t) = 1$ if $s = t$ and $= 0$ otherwise; (ii) Union $(R \cup Q)(s,t) = R(s,t) \vee^{\boldsymbol{Ł}_n} Q(s,t)$; (iii) Composition $(R \circ Q)(s,t) = \bigvee^{\boldsymbol{Ł}_n} \{R(s,u) \odot Q(u,t) \mid u \in S\}$; (iv) Finite iteration $R^0 = \iota$, $R^{n+1} = R^n \circ R$; (v) Transitive closure $R^+ = \bigvee^{\boldsymbol{Ł}_n} \{R^n \mid n > 0\}$; (vi) Reflexive transitive closure $R^* = \bigvee^{\boldsymbol{Ł}_n} \{R^n \mid n \geq 0\}$. (Note that arbitrary unions are well defined since $\boldsymbol{Ł}_n$ is finite.) Alternatively, $R^+(s,t) = \bigvee_{\sigma \in S^*} Rs\sigma t$, where S^* is the set of all finite sequences of elements of S, including the empty sequence $\emptyset$, and where $Rs\sigma t$ is defined by induction on the length of σ by setting $Rs\emptyset t := R(s,t)$ and $Rs(\sigma^\frown u)t := Rs\sigma u \odot R(u,t)$, where $\sigma^\frown u$ is the result of appending u to the end of sequence σ. It is clear that $R^*(s,t) = \iota(s,t) \vee^{\boldsymbol{Ł}_n} R^+(s,t)$.

Let T, S be two sets. We lift the $\boldsymbol{Ł}_n$ operations to functions $f, g, \ldots$ from $T \times S$ to $\boldsymbol{Ł}_n$ by defining $(f \oplus g)(t,s) = f(t,s) \oplus g(t,s)$, $(f \odot g)(t,s) = f(t,s) \odot g(t,s)$, $(\sim f)(t,s) = \sim f(t,s)$, $1(t,s) = 1$ and $0(t,s) = 0$.

3 Propositional Dynamic Logic Over $\boldsymbol{Ł}_n$

Let A be a countable set of program variables $a_0, a_1, \ldots$. Let $\boldsymbol{Ł}_n$ be a finite Łukasiewicz chain. The sets of *programs* Π_n and *formulas* Φ_n of the *dynamic language* $\mathcal{L}_n$ are defined by mutual induction as follows:

- Π_n $\quad \alpha := a \mid \alpha \cup \alpha \mid \alpha ; \alpha \mid \alpha^* \mid \varphi?$
- Φ_n $\quad \varphi := p \mid \bar{c} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid [\alpha]\varphi$

where $a \in A$, $p \in P$ and $c \in Ł_n$. We apply to $\mathcal{L}_n$ all the definitions and conventions adopted for $\mathcal{P}_n$; moreover, we define $\langle\alpha\rangle\varphi := \neg[\alpha]\neg\varphi$. An $\mathcal{L}_n$*-expression* is any element of $\Pi_n \cup \Phi_n$.

Definition 3.1. A $\boldsymbol{Ł}_n$-*frame* is $\mathfrak{F} = (S, E)$, where $S \neq \emptyset$ and E is a function from $\mathbb{N}$ to $\boldsymbol{Ł}_n$-valued binary relations on S. A $\boldsymbol{Ł}_n$-*model* based on $\mathfrak{F} = (S, E)$ is $\mathfrak{M} = (S, E, V)$, where V is a function from $\mathbb{N}$ to functions from S to $\boldsymbol{Ł}_n$. For any $\mathfrak{M}$, the $\mathfrak{M}$-*interpretation function* $I_{\mathfrak{M}}$ is a function $\big((\Pi_n \times S \times S) \cup (\Phi_n \times S)\big) \to \boldsymbol{Ł}_n$ such that (R_α is a $\boldsymbol{Ł}_n$-valued binary relation defined by $R_\alpha(s,t) := I_{\mathfrak{M}}(\alpha, s, t)$)

- $I_{\mathfrak{M}}(p_i, s) = V(i)(s)$;
- $I_{\mathfrak{M}}(\bar{c}, s) = c$;
- $I_{\mathfrak{M}}(\varphi \star \psi, s) = I_{\mathfrak{M}}(\varphi, s) \star^{\boldsymbol{Ł}_n} I_{\mathfrak{M}}(\psi, s)$ for $\star \in \{\wedge, \vee, \to\}$;
- $I_{\mathfrak{M}}([\alpha]\varphi, s) = \bigwedge^{\boldsymbol{Ł}_n}_{t \in S} \big(I_{\mathfrak{M}}(\alpha, s, t) \to^{\boldsymbol{Ł}_n} I_{\mathfrak{M}}(\varphi, t)\big)$;
- $I_{\mathfrak{M}}(a_i, s, t) = E(i)(s, t)$;
- $I_{\mathfrak{M}}(\alpha \cup \beta, s, t) = (R_\alpha \cup R_\beta)(s, t)$;
- $I_{\mathfrak{M}}(\alpha; \beta, s, t) = (R_\alpha \circ R_\beta)(s, t)$;
- $I_{\mathfrak{M}}(\alpha^*, s, t) = (R_\alpha)^*(s, t)$
- $I_{\mathfrak{M}}(\varphi?, s, t) = \begin{cases} I_{\mathfrak{M}}(\varphi, s) & \text{if } s = t; \\ 0 & \text{otherwise.} \end{cases}$

A formula φ is *valid in* $\mathfrak{M}$ iff $I_{\mathfrak{M}}(\varphi, s) = 1$ for all $s \in S$; a formula φ is *valid in* $\mathfrak{F}$ iff it is valid in each $\mathfrak{M}$ based on $\mathfrak{F}$. **PDŁ**$_n \subseteq \Phi_n$ is the set of formulas valid in all $\boldsymbol{Ł}_n$-frames. A formula φ *is satisfiable* in a class of frames iff $I_{\mathfrak{M}}(\varphi, s) > 0$ for some s in $\mathfrak{M}$ based on a $\mathfrak{F}$ in the class; that is, iff $\neg\varphi$ is not valid in the class.

If $\mathfrak{M}$ is understood from the context, we will often write $\alpha/_{st}$ instead of $I_{\mathfrak{M}}(\alpha, s, t)$ and $\varphi/_s$ instead of $I_{\mathfrak{M}}(\varphi, s)$.

Note that we have the following in all $\mathfrak{M}$:

$$\begin{aligned} I_{\mathfrak{M}}(\langle\alpha\rangle\varphi, s) &= 1 - \min\left\{\alpha/_{st} \to^{\boldsymbol{Ł}_n} (1 - \varphi/_t) \mid t \in S\right\} \\ &= 1 - \min\{\sim\alpha/_{st} \oplus \sim\varphi/_t \mid t \in S\} \\ &= \max\{\sim(\sim\alpha/_{st} \oplus \sim\varphi/_t) \mid t \in S\} \\ &= \max\{\alpha/_{st} \odot \varphi/_t \mid t \in S\} \end{aligned}$$

4 A Practical Example

In this section we discuss the informal interpretation **PDŁ**$_n$ in more detail. We also hint at possible knowledge representation applications by means of a practical example.

The value of $\alpha/_{st}$ may be seen as the truth degree of the statement "t is easily accessible from s by α" where "easily accessible" is understood as a graded relation. This means that $1 - \alpha/_{st}$ can be seen as the cost, as discussed in Section 2.1, of accessing t from s via α. If this cost is absolute, i.e. equal to 1, then we may say that t is not accessible from s via α; if it is equal to 0, then we may say that t is accessible from s via α for free.

Example 4.1. Imagine a robot in the starting position s to the left of a row of three tables, with boxes on top of them; see Figure 4.1. Assume that the robot needs 10 seconds to move between s and the first table, t_1, by performing the action right, and similarly for moving further right to the other tables. With a particular time-frame in mind, this could be modelled by associating a cost of 0.1 with each right transition, depicted as a right-pointing arrow in the picture; note that if the cost is 0.1, then the truth degree of "the robot can access ... from ... easily by moving right"—where "easily" may be understood as "quickly"—is 0.9. Assume furthermore that the box on top of table 1 weights 10 kg, the one on top of table 2 weights 15 kg and the one on top of table 3 weights 5 kg. With some particular capacity of the robot in mind, this could be modelled by varying truth degrees assigned to the propositional variable Heavy, representing the graded statement that there is a heavy box on the table in the given position, for instance as indicated in the picture by the numbers below the states in the bottom row (note that there is no table in the starting position, so the truth degree of Heavy in s is 0).[5] The upward pointing arrows represent the action pick_up of picking up the box from the table (more precisely, the truth degree of "the robot can get easily from ... to ... by picking up the box"); note that the cost depends on the weight of the box. The left-pointing arrows and the arrow from u_1 to s represent the action left of moving left; note that the cost associated with each left transition is bigger than the cost associated with right transitions since we assume that the robot moves left while carrying a box. Note also that the truth degree of Heavy in the u_i states is 0 since the box is carried by the robot. (Here we simplify the model by assuming that only one box gets picked up; however, the values of Heavy at u_i will not matter in our further discussion.) Assume furthermore that we have a propositional variable End, whose truth degree is 1 at s and 0 at all the other states. (The example is heavily idealised and many possible transitions are left out for the sake of simplicity.)

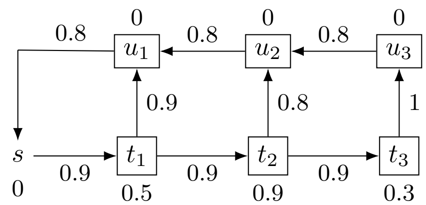

Figure 1: A box world example.

Let us now discuss the interpretation of modal formulas, before using them to express various interesting features of Example 4.1. Take any states s, t in an arbitrary model $\mathfrak{M}$, any program α and formula φ. The value of $\alpha/_{st} \to^{\boldsymbol{Ł}_n} \varphi/_t$ is equal to $\sim(\alpha/_{st}) \oplus \varphi/_t$, the (truncated) sum of the cost of accessing t from s by α and the truth degree of φ in t. Hence, $\alpha/_{st} \to^{\boldsymbol{Ł}_n} \sim(\varphi/_t)$ is $\sim(\alpha/_{st}) \oplus \sim(\varphi/_t)$, which we may understand as the *cost sum* of accessing t from s via α and obtaining φ at t. This means that the value of $[\alpha]\neg\varphi$ at s is *the minimal cost sum of obtaining* φ *via* α. Let us write MinCost(α, φ) instead of $[\alpha]\neg\varphi$. (Because of space limitations, we will not discuss interpretations of other combinations of modality and propositional connectives and we will focus only on the most intuitive one.)

[5]Since our language is propositional, we cannot express statements about the weight of objects on tables in t_i in a non-indexical way, e.g. as Heavy(object_on_table(t_i)). Extensions of the present framework to fragments of first-order dynamic logic are a natural topic of future work.

Returning to Example 4.1, consider the Kleene star of right, representing the action of *moving right some finite number of steps*. We have (taking Ł_{11} as our algebra of truth degrees):

v	s	t_1	t_2	t_3	u_i
$\mathsf{right}^*/_{sv}$	1	0.9	0.8	0.7	0
$\mathsf{Heavy}/_v$	0	0.5	0.9	0.3	0
$\sim(\mathsf{right}^*/_{sv})\oplus \sim(\mathsf{Heavy}/_v)$	1	0.6	0.3	1	1

and so $\mathsf{MinCost}(\mathsf{right}^*, \mathsf{Heavy})/_s = 0.3$. Hence, the minimal cost the robot needs to "pay" when the goal is "*to get quickly to a table with a heavy box on top of it*", and the space of actions considered is *to move right some finite number of steps*, is 0.3.

The MinCost construction allows to express information that is crucial in deciding between various courses of action. The implication connective adds to this expressivity explicit means of comparison, as indicated by the following. Let φ_n stand for $\mathsf{MinCost}(\mathsf{right}^n, \mathsf{Heavy})$ and recall that $x \to^{\text{Ł}_{11}} y = 1$ if $x \sqsubseteq y$ and otherwise $x \to^{\text{Ł}_{11}} y = 1 - x + y$. For instance, the value of

$$\mathsf{MinCost}(\mathsf{right}, \mathsf{Heavy}) \to \mathsf{MinCost}(\mathsf{right}^*, \mathsf{Heavy})$$

at s is $0.6 \to^{\text{Ł}_{11}} 0.3 = 0.7$, indicating that the robot can do better than moving right one step, i.e. that one is not the optimal number of steps. However, the truth degree 0.7 carries finer-grained information, namely, *how close* to the optimum taking one step is. Note that the value of

$$\mathsf{MinCost}(\mathsf{right}^3, \mathsf{Heavy}) \to \mathsf{MinCost}(\mathsf{right}^*, \mathsf{Heavy})$$

at s is $1 \to^{\text{Ł}_{11}} 0.3 = 0.3$; that is, both right and right^3 are not optimal, but right is closer.

As a more complex example, consider

$$\alpha = \mathsf{right}^* ; \mathsf{pick\_up} ; \mathsf{left}^*$$

expressing the action of *moving a finite number of steps to the right, then picking up a box from the table, and then moving a finite number of steps to the left*. We have $\mathsf{MinCost}(\alpha, \mathsf{End})/_s = 0.4$, the "optimal instance" being right ; pick_up ; left. Hence, if the goal is to "*get quickly to a box, pick it up easily and get back quickly*", then the optimal action is to go via t_1. However, if the goal takes into account also the weight of the retrieved box, say "*get quickly to a heavy box, pick it up easily and get back quickly*", then the situation changes. Consider the program

$$\beta = \mathsf{right}^* ; \mathsf{Heavy}? ; \mathsf{pick\_up} ; \mathsf{left}^* .$$

It is easily checked that $\mathsf{MinCost}(\beta, \mathsf{End})/_s = 0.9$ but now the instances $\beta_1 = \mathsf{right} ; \mathsf{Heavy}? ; \mathsf{pick\_up} ; \mathsf{left}$ and $\beta_2 = \mathsf{right}^2 ; \mathsf{Heavy}? ; \mathsf{pick\_up} ; \mathsf{left}^2$ are equally optimal, as the values of both

$$\mathsf{MinCost}(\beta_1, \mathsf{End}) \to \mathsf{MinCost}(\beta, \mathsf{End})$$
$$\mathsf{MinCost}(\beta_2, \mathsf{End}) \to \mathsf{MinCost}(\beta, \mathsf{End})$$

at s are 1.

Finally, let us remark that truth-degree constants $\bar{c}$ for $c \in \text{Ł}_n$ are useful in expressing the degrees to which some costs exceed specific thresholds. In Example 4.1 we have

$$\big(\mathsf{MinCost}(\alpha, \mathsf{End}) \to \overline{0.5}\big)/_s = 1$$
$$\big(\mathsf{MinCost}(\beta, \mathsf{End}) \to \overline{0.5}\big)/_s = 0.6$$

One way to read formulas of the form $\mathsf{MinCost}(\alpha, \varphi) \to \bar{c}$ is "The minimal cost of accessing φ via α does not exceed the threshold c much". In many settings the finer-grained information specifying *how much* a threshold is exceeded is more useful than information *whether* the threshold is exceeded.

5 Completeness and Decidability

In this section we establish that, for each $n > 1$, the validity problem for $\mathbf{PDŁ}_n$ is decidable (Theorem 5.12); this result is obtained via a recursive axiomatization result for each $\mathbf{PDŁ}_n$ (Theorem 5.11) and proving that each $\mathbf{PDŁ}_n$ has the finite model property—if φ is not valid in $\mathbf{PDŁ}_n$, then there is a finite model invalidating ϕ (Theorem 5.10).

Definition 5.1. For each $n \geq 2$, let $PDŁ_n$ be the Hilbert-style proof system that extends Ł_n with axiom schemata

(A1) $[\,\alpha\,]\,\bar{1}$

(A2) $[\,\alpha\,]\,\varphi \wedge [\,\alpha\,]\,\psi \to [\,\alpha\,]\,(\varphi \wedge \psi)$

(A3) $[\,\alpha\,]\,(\bar{c} \to \varphi) \leftrightarrow (\bar{c} \to [\,\alpha\,]\,\varphi)$

(A4) $[\,\alpha \cup \beta\,]\,\varphi \leftrightarrow ([\,\alpha\,]\,\varphi \wedge [\,\beta\,]\,\varphi)$

(A5) $[\,\alpha;\beta\,]\,\varphi \leftrightarrow [\,\alpha\,]\,[\,\beta\,]\,\varphi$

(A6) $[\,\alpha^*\,]\,\varphi \leftrightarrow (\varphi \wedge [\,\alpha\,]\,[\,\alpha^*\,]\,\varphi)$

(A7) $[\,\varphi?\,]\,\psi \leftrightarrow (\varphi \to \psi)$

and rules

(R1) $\dfrac{\varphi \to \psi}{[\,\alpha\,]\,\varphi \to [\,\alpha\,]\,\psi}$ (R2) $\dfrac{\varphi \to [\,\alpha\,]\,\varphi}{\varphi \to [\,\alpha^*\,]\,\varphi}$

(A1–3) and (R1) come from the axiomatization of modal logics over finite residuated lattices (Bou et al. 2011), the rest is the standard axiomatization of crisp **PDL**.

Let f be a Ł_n-valuation of Φ_n; f is called $PDŁ_n$-*correct* (or just n-correct) iff $f(\varphi) = 1$ for all theorems φ of $PDŁ_n$. The following lemma shows that all theorems are valid.

Lemma 5.2. *For each Ł_n-model $\mathfrak{M}$ and each s in $\mathfrak{M}$, the function e_s defined by $e_s(\varphi) = I_{\mathfrak{M}}(\varphi, s)$, is an n-correct Ł_n-valuation.*

Proof. The function e_s is a Ł_n-valuation by definition. The fact that it is n-correct is established by routine induction on the length of proofs. The base case follows easily from the definitions (e.g. clearly $R_{\alpha^*}st = \iota(s,t) \vee^{\text{Ł}_n} (R_\alpha \circ R_{\alpha^*})st)$. We prove that (R2) preserves validity; the argument is analogous to the one that applies in the crisp case. Assume that $\varphi/_s \sqsubseteq [\,\alpha\,]\,\varphi/_s$ for all s in some fixed $\mathfrak{M}$. Clearly $\varphi/_u \sqsubseteq \iota(u,t) \to^{\text{Ł}_n} \varphi/_t$ for all u and t. Moreover, it can be shown by easy induction on the length of finite sequences of states in $\mathfrak{M}$, that $\varphi/_u \sqsubseteq (R_\alpha)^+ st \to^{\text{Ł}_n} \varphi/_t$ for all t. Hence, $\varphi/_u \sqsubseteq \min\{(\iota(u,t) \vee^{\text{Ł}_n} (R_\alpha)^+ ut) \to^{\text{Ł}_n} \varphi/_t\}$, which means that $(\varphi \to [\,\alpha^*\,]\,\varphi)/_u = 1$ for any u in $\mathfrak{M}$. □

Definition 5.3. For any n, the structure $\mathfrak{I}_n = (S_n, I_n)$ consists of the set S_n of all n-correct valuations and a function I_n such that $I_n(\varphi, s) = s(\varphi)$ for all $s \in S_n$ and $I_n(\alpha, s, t) = \bigwedge^{\text{Ł}_n}_{\varphi \in \Phi_n} \{s([\alpha]\varphi) \to^{\text{Ł}_n} t(\varphi)\}$. The *canonical PDŁ$_n$-model* is $\mathfrak{M}_n = (S_n, E_n, V_n)$ where $E_n(i)(s,t) = I_n(a_i, s, t)$ and $V_n(i)(s) = I_n(p_i, s)$.

As witnessed already in the classical case $n = 2$, properties of the Kleene star operator preclude us from establishing that $I_n(\varphi, s) = I_{\mathfrak{M}_n}(\varphi, s)$ for all $\varphi \in \Phi_n$ and thus to prove completeness of $PDŁ_n$ with respect to Ł_n-frames using the canonical model technique. As is well-known for $n = 2$, completeness can be established using filtration of the canonical model. We show here that the filtration argument can be generalized to arbitrary $n \in \mathbb{N}$. (Filtrations in the context of many-valued logic were studied previously in (Conradie, Morton, and Robinson 2017).)

A set of formulas Γ is *Fischer-Ladner closed* if (i) it contains all subformulas of all $\varphi \in \Gamma$; (ii) it contains $[\alpha]\varphi$ and $[\beta]\varphi$ if $[\alpha \cup \beta]\varphi \in \Gamma$; (iii) it contains $[\alpha][\beta]\varphi$ if $[\alpha;\beta]\varphi \in \Gamma$; (iv) it contains $[\alpha][\alpha^*]\varphi$ if $[\alpha^*]\varphi \in \Gamma$; (v) it contains $\psi \to \varphi$ if $[\psi?]\varphi \in \Gamma$. The *Fischer-Ladner closure* of Γ, $FL(\Gamma)$, is the smallest Fischer-Ladner closed set containing Γ as a subset; we write $FL(\varphi)$ for $FL(\{\varphi\})$.

For any $s, t \in S_n$, we define

$$s \approx^\Gamma t \iff \forall \varphi \in \Gamma (s(\varphi) = t(\varphi))$$

(Hence, $\approx^\Gamma$ is a crisp (two-valued) relation on S_n.) We denote the equivalence class of s under $\approx^\Gamma$ as $|s|_\Gamma$.

Definition 5.4. Let Γ be an arbitrary finite Fischer-Ladner closed subset of Φ_n. The *filtration of* $\mathfrak{M}_n$ *through* Γ is $\mathfrak{M}_n^\Gamma = (S_n^\Gamma, E_n^\Gamma, V_n^\Gamma)$, where

- $S_n^\Gamma = \{|s|_\Gamma \mid s \in S_n\}$
- $E_n^\Gamma(i)(|s|_\Gamma, |t|_\Gamma) =$ $$\bigwedge^{\text{Ł}_n}\{s([a_i]\varphi) \to^{\text{Ł}_n} t(\varphi) \mid [a_i]\varphi \in \Gamma\}$$
- $V_n^\Gamma(i)(s) = V_n(i)(s)$ if $p_i \in \Gamma$ and $= 0$ otherwise.

$I_{\mathfrak{M}_n^\Gamma}$ is defined as for Ł_n-models. We will write I_n instead of $I_{\mathfrak{M}_n}$ and I_n^Γ instead of $I_{\mathfrak{M}_n^\Gamma}$. When Γ is clear from the context, we will usually write $|s|$ instead of $|s|_\Gamma$.

$\mathfrak{M}_n^\Gamma$ is a Ł_n-model by definition. To prove completeness, we need to show that, for all $\varphi \in \Gamma$ and all $s \in S_n$, $I_n(\varphi, s) = I_n^\Gamma(\varphi, |s|)$.

In what follows, we write $R_\alpha st$ instead of $I_n(\alpha, s, t)$ and $R_\alpha^\Gamma|s||t|$ instead of $I_n^\Gamma(\alpha, |s|, |t|)$.

In proofs by induction on the complexity of expressions, we use the following complexity measure Co:

(i) $Co(p) = Co(a) = 0$;
(ii) $Co(\bar{c}) = 1$;
(iii) $Co(\varphi \star \psi) = Co(\varphi) + Co(\psi) + 1$, for $\star \in \{\wedge, \vee, \to\}$;
(iv) $Co([\alpha]\varphi) = Co(\alpha) + Co(\varphi) + 1$
(v) $Co(\alpha \cup \beta) = Co(\alpha) + Co(\beta) + 1$;
(vi) $Co(\alpha;\beta) = Co(\alpha) + Co(\beta) + 1$;
(vii) $Co(\alpha^*) = Co(\alpha) + 1$;
(viii) $Co(\varphi?) = Co(\varphi) + 1$.

Lemma 5.5. *The following holds for all $n \in \mathbb{N}$, all $\alpha \in \Pi_n$, and all $s \in S_n$:*

1. *For all $\varphi \in \Phi_n$, $s([\alpha]\varphi) = \bigwedge^{\text{Ł}_n}_{u \in S_n} \{R_\alpha su \to^{\text{Ł}_n} u(\varphi)\}$.*
2. *For all $t \in S_n$, $R_\alpha st = \bigwedge^{\text{Ł}_n}_{\psi \in \Phi_n} \{t(\psi) \mid s([\alpha]\psi) = 1\}$.*

Proof. This follows from the work of (Bou et al. 2011) on modal logics over finite residuated lattices; see Lemma 4.8 (the first claim) and Prop. 4.1 (the second claim) in (Bou et al. 2011). The proofs given in that paper apply to our case since Ł_n is finite and we have a strongly complete axiomatization of Ł_n-consequence by Theorem 2.6. ☐

Definition 5.6. For all n, $\alpha \in \Pi_n$ and $s \in S_n$, and for all finite closed sets Γ, we define

$$R_\alpha^{n,\Gamma} s := \bigvee_{x \in S_n} \left(\overline{R_\alpha^\Gamma|s||x|} \;\&\; \bigwedge_{\varphi \in \Gamma} \left(\overline{x(\varphi)} \leftrightarrow \varphi\right)^n \right).$$

If n and Γ are clear from the context, then we write just $R_\alpha s$.

Lemma 5.7. *Fix any n, a finite Fischer-Ladner closed Γ and $s \in S_n$. Then, for all $t \in S_n$, $t(R_\alpha s) = R_\alpha^\Gamma|s||t|$.*

Proof. Clearly

$$t(R_\alpha s) = \bigvee^{\text{Ł}_n}_{x \in S_n} \left\{ R_\alpha^\Gamma|s||x| \odot \bigwedge^{\text{Ł}_n}_{\varphi \in \Gamma} \left(x(\varphi) \leftrightarrow^{\text{Ł}_n} t(\varphi)\right)^n \right\}.$$

By Lemma 2.3, this equals $\bigvee^{\text{Ł}_n}_{x \in S_n}\{R_\alpha^\Gamma|s||x| \mid x \approx^\Gamma t\} = R_\alpha^\Gamma|s||t|$. ☐

Lemma 5.8. $s([\alpha]R_\alpha s) = 1$ *iff* $(\forall t \in S_n)(R_\alpha st \sqsubseteq R_\alpha^\Gamma st)$.

Proof. $s([\alpha]R_\alpha s) = \bigwedge^{\text{Ł}_n}_{t \in S_n}\{R_\alpha st \to^{\text{Ł}_n} t(R_\alpha s)\}$ by Lemma 5.5 and so $s([\alpha]R_\alpha s) = \bigwedge^{\text{Ł}_n}_{t \in S_n}\{R_\alpha st \to^{\text{Ł}_n} R_\alpha^\Gamma|s||t|\}$ by Lemma 5.7. This means that $s([\alpha]R_\alpha s) = 1$ iff $R_\alpha st \sqsubseteq R_\alpha^\Gamma|s||t|$ as required. ☐

Lemma 5.9. $s(\varphi) = 1$ *for all $s \in S_n$ iff φ is a theorem of $PDŁ_n$.*

Proof. The "if" implication holds by definition. Conversely, if $\nvdash_{PDŁ_n} \varphi$, then $T \nvdash_{\text{Ł}_n} \varphi$, where T is the set of theorems of $PDŁ_n$. (Otherwise $T \vdash_{PDŁ_n} \varphi$ and so $\vdash_{PDŁ_n} \varphi$.) By Theorem 2.6, there is a Ł_n-homomorphism h such that $h(\psi) = 1$ for all $\psi \in T$ and $h(\varphi) \neq 1$. Hence, $h \in S_n$. ☐

Theorem 5.10 (Filtration)**.** *The following holds for all $n \in \mathbb{N}$ and all finite closed $\Gamma \subseteq \Phi_n$:*

1. *If $\varphi \in \Gamma$, then $I_n(\varphi, s) = I_n^\Gamma(\varphi, |s|)$.*
2. $R_\alpha st \sqsubseteq R_\alpha^\Gamma|s||t|$.
3. *If $[\alpha]\varphi \in \Gamma$, then $s([\alpha]\varphi) \sqsubseteq R_\alpha^\Gamma|s||t| \to^{\text{Ł}_n} t(\varphi)$.*

Proof. Simultaneous induction on the complexity of expressions, similar to the proof in the two-valued case.

The base cases follow easily from the definitions. The cases of the induction step of the first claim dealing with non-modal connectives are easy. For instance, $I_n(\bar{c}, s) = s(\bar{c}) = c = I_n^\Gamma(\bar{c}, s)$. The case of $\varphi = [\alpha]\psi$ follows from the second and third claim of the lemma (which can be used since $Co(\alpha) < Co([\alpha]\psi)$) and Lemma 5.5.

Now we establish the induction step of the second claim. Assume that $\alpha = \beta \cup \gamma$. Then $s([\delta] R_\delta s) = 1$ for $\delta \in \{\beta, \gamma\}$ by the induction hypothesis and Lemma 5.8. By Lemma 5.5, $s([\delta](R_\beta s \vee R_\gamma s))$. By the definition of $R^\Gamma_{\beta\cup\gamma}$ and Lemma 5.7, $t(R_{\beta\cup\gamma}s) = t(R_\beta s \vee R_\gamma s)$ and so $s([\delta] R_{\beta\cup\gamma}s) = 1$. But this means that $s([\beta \cup \gamma] R_{\beta\cup\gamma}s) = 1$, which entails $R_{\beta\cup\gamma}st \sqsubseteq R^\Gamma_{\beta\cup\gamma}|s||t|$ by Lemma 5.8.

Let $\alpha = \beta;\gamma$. Then, for all $u, v \in S_n$, $R_\beta su \odot R_\gamma uv \sqsubseteq R^\Gamma_{\beta;\gamma}|s||v|$ by the induction hypothesis and the definition of $R^\Gamma_{\beta;\gamma}$. By Lemma 5.7, this entails that $R_\beta su \sqsubseteq \bigwedge^{Ł_n}_{w\in S_n}\{R_\gamma uw \to^{Ł_n} w(R_{\beta;\gamma}s)\}$ and so, using Lemma 5.5, $1 = \big(R_\beta su \to^{Ł_n} u([\gamma] R_{\beta;\gamma}s)\big)$, which means, since u is arbitrary, that $1 = s([\beta][\gamma] R_{\beta;\gamma}s) = s([\beta;\gamma] R_{\beta;\gamma}s)$. Hence, $R_{\beta;\gamma}st \sqsubseteq R^\Gamma_{\beta;\gamma}|s||t|$ for any t by Lemma 5.8.

Let $\alpha = \beta^*$. It follows from the definition of $R^\Gamma_{\beta^*}$ that $R^\Gamma_{\beta^*}|s||u| \odot R^\Gamma_\beta|u||t| \sqsubseteq R^\Gamma_{\beta^*}|s||t|$, for all $s, t, u \in S_n$. By the induction hypothesis and Lemma 5.7, $u(R_{\beta^*}s) \odot R_\beta ut \sqsubseteq t(R_{\beta^*}s)$. Since u is arbitrary, it follows that $u(R_{\beta^*}s \to [\beta] R_{\beta^*}s) = 1$ by Lemma 5.5 and so $\vdash_{PDŁ_n} R_{\beta^*}s \to [\beta] R_{\beta^*}s$ by Lemma 5.9. Using (R2), $\vdash_{PDŁ_n} R_{\beta^*}s \to [\beta^*] R_{\beta^*}s$, which means that $s(R_{\beta^*}s) \sqsubseteq s([\beta^*] R_{\beta^*}s)$. But $s(R_{\beta^*}s) = 1$ by Lemma 5.7 and the definition of R_{β^*}. Hence, $s([\beta^*] R_{\beta^*}s) = 1$ and so $R_{\beta^*}st \sqsubseteq R^\Gamma_{\beta^*}|s||t|$ by Lemma 5.8.

Let $\alpha = \psi?$. By the definition of $R^\Gamma_{\psi?}$, $I_\Gamma(\psi, |s|) \sqsubseteq R^\Gamma_{\psi?}|s||s|$. Hence, $s(\psi) \sqsubseteq s(R_{\psi?}s)$ by Lemma 5.7 and the induction hypothesis (which may be used since $Co(\psi) < Co(\psi?)$). It follows that $s(\psi \to R_{\psi?}s) = 1$ which, using (A7), entails $s([\psi?] R_{\psi?}s)$. Hence, $R_{\psi?}st \sqsubseteq R^\Gamma_{\psi?}|s||t|$ by Lemma 5.8.

Now we establish the induction step of the third claim. Let $\alpha = \beta \cup \gamma$. If $x \sqsubseteq s([\beta \cup \gamma]\varphi)$, then $x \sqsubseteq s([\delta]\varphi)$ for $\delta\{\beta, \gamma\}$ thanks to (A4). By the induction hypothesis and the properties of Γ, $x \sqsubseteq R^\Gamma_\delta|s||t| \to^{Ł_n} t(\varphi)$ which, by definition of $R^\Gamma_{\beta\cup\gamma}$ entails that $x \sqsubseteq R_{\beta\cup\gamma}|s||t| \to^{Ł_n} t(\varphi)$.

Let $\alpha = \beta;\gamma$. If $x \sqsubseteq s([\beta;\gamma]\varphi)$, then $x \sqsubseteq s([\beta][\gamma]\varphi)$ thanks to (A5). By the induction hypothesis and the properties of Γ, $x \sqsubseteq R^\Gamma_\beta|s||u| \to^{Ł_n} (R^\Gamma_\gamma|u||t| \to^{Ł_n} t(\varphi))$ for any u. Hence, $x \sqsubseteq R^\Gamma_{\beta;\beta}|s||t| \to^{Ł_n} t(\varphi)$ by definition of $R_{\beta;\gamma}$.

Let $\alpha = \beta^*$. If $x \sqsubseteq s([\beta^*]\varphi)$, then $x \sqsubseteq s(\varphi \wedge [\beta][\beta^*]\varphi)$. Hence, $x \sqsubseteq s(\varphi)$ and so $x \sqsubseteq R^\Gamma_{\beta^*}|s||s| \to^{Ł_n} s(\varphi)$ by the definition of $R^\Gamma_{\beta^*}$. Now assume that $|s| \neq |t|$. Hence, $R^\Gamma_{\beta^*}|s||t| = (R^\Gamma_\beta)^+|s||t| = \bigvee^{Ł_n}_{\sigma\in(S^\Gamma_n)^*}(R_\beta|s|\sigma|t|)$. We prove that $x \sqsubseteq R_\beta|s|\sigma|u| \to^{Ł_n} t([\beta^*]\varphi)$ for all u and all $\sigma \in (S^\Gamma_n)^*$ by induction on the length of σ. If follows from this using (A6) that $x \sqsubseteq (R^\Gamma_\beta)^+|s||t| \to^{Ł_n} t(\varphi)$. Now if $\sigma = \emptyset$, then we reason as follows. We know that $x \sqsubseteq s([\beta][\beta^*]\varphi)$ and so, by the induction hypothesis, $x \sqsubseteq R^\Gamma_\beta|s||t| \to^{Ł_n} t([\beta^*]\varphi)$, which means that $x \sqsubseteq R^\Gamma_\beta|s|\emptyset|t| \to^{Ł_n} t(\varphi)$. Now assume that $x \sqsubseteq R^\Gamma_\beta|s|\rho|u| \to^{Ł_n} t([\beta^*]\varphi)$. Using (A6) we obtain $x \sqsubseteq R^\Gamma_\beta|s|\rho|u| \to^{Ł_n} t([\beta][\beta^*]\varphi)$ and so, by the induction hypothesis (of the third claim of the theorem), $x \sqsubseteq R^\Gamma_\beta|s|\rho|u| \to^{Ł_n} (R^\Gamma_\beta|u||t| \to^{Ł_n} t([\beta^*]\varphi))$. It follows that $x \sqsubseteq R_\beta|s|(\rho^\frown u)|t| \to^{Ł_n} t([\beta^*] f)$.

Finally, let $\alpha = \psi?$. If $x \sqsubseteq ([\psi?]\varphi)$, then $x \sqsubseteq s(\psi) \to^{Ł_n} s(\varphi)$, which means that $x \sqsubseteq R^\Gamma_{\psi?}|s||s| \to^{Ł_n} s(\varphi)$ by the induction hypothesis and the definition of R^Γ_ψ. Now take t such that $|s| \neq |t|$. We know that $(0 \to^{Ł_n} x) = 1$ and so clearly $x \sqsubseteq R_{\psi?}|s||t| \to^{Ł_n} t(\varphi)$. □

Theorem 5.11 (Completeness). *For all $n > 1$, **PDŁ**$_n$ is the set of theorems of $PDŁ_n$.*

Proof. Soundness follows from Lemma 5.2. Completeness follows from Lemma 5.9 and Theorem 5.10 (for each non-theorem φ, consider $\mathfrak{M}^\Gamma_n$ where Γ is the closure of $\{\varphi\}$). □

Theorem 5.12 (Decidability). *For all $n > 1$, **PDŁ**$_n$ is a decidable set.*

Proof. **PDŁ**$_n$ is recursively axiomatizable by Theorem 5.11 and it has the finite model property by Theorem 5.10—if φ is not valid in some model $\mathfrak{M}$, then it is not valid in the finite model $\mathfrak{M}^{FL(\varphi)}$. □

6 Complexity

In this section we establish the complexity of the validity problem for each **PDŁ**$_n$; it is shown that, in each case, the problem is $EXPTIME$-complete.

6.1 A Deterministic Exponential-time Algorithm

We generalize Pratt's deterministic exponential-time algorithm (Pratt 1979; Harel, Kozen, and Tiuryn 2000) for checking satisfiability in **PDL** (= **PDŁ**$_2$) to **PDŁ**$_n$ for arbitrary $n > 1$.

Fix some $n > 1$ and $\chi \in \Phi_n$. Let $\Gamma = FL(\chi)$. For $|s|_\Gamma \in S^\Gamma_n$, we define a function $f_{|s|_\Gamma} : \Gamma \to Ł_n$ by $f_{|s|_\Gamma}(\varphi) = s(\varphi)$; we often write f_s instead of $f_{|s|_\Gamma}$. Let $N := \{f_s \mid |s|_\Gamma \in S^\Gamma_n\}$. Taking N as the universe, define the model $\mathfrak{N} = (N, E_\mathfrak{N}, V_\mathfrak{N})$ exactly as $\mathfrak{M}^\Gamma_n$; that is, $E_\mathfrak{N}(j)(h, g) = \bigwedge^{Ł_n}\{h([a_j]\varphi) \to^{Ł_n} g(\varphi) \mid [a_j]\varphi \in \Gamma\}$ and $V_\mathfrak{N}(j)(h) = h(p_j)$ if $p_j \in \Gamma$ and $= 0$ otherwise. Let $I_\mathfrak{N}$ be defined as usual. It is clear that $\mathfrak{N}$ is isomorphic to $\mathfrak{M}^{FL(\chi)}_n$. The question is how to generate $\mathfrak{N}$ without going via the (uncountable) canonical model $\mathfrak{M}_n$. This is the job of Algorithm 1 shown on page 8.

The following is a generalization of Lemma 8.3 in (Harel, Kozen, and Tiuryn 2000).

Lemma 6.1. *Assume that $N \subseteq M$, with E and I defined as in Part 3 of Algorithm 1. Let $\theta \in \Gamma$ such that, for all $[\alpha]\varphi \in FL(\theta)$ and all $h \in M$, the condition* (1) *is satisfied. Then:*

1. *For all $\psi \in FL(\theta)$ and $f \in M$, $h(\psi) = I(\psi, h)$*
2. *For all $[\alpha]\psi \in FL(\theta)$ and all $h, g \in M$:*
   (a) $I_\mathfrak{N}(\alpha, h, g) \sqsubseteq I(\alpha, h, g)$;
   (b) $h([\alpha]\psi) \sqsubseteq I(\alpha, h, g) \to^{Ł_n} g(\psi)$.

Proof. Simultaneous induction on the complexity of expressions, with θ fixed. The base case of claim 1 holds by definition of V. The non-modal claims of the induction step are established easily using the induction hypothesis and the fact that $h \in M_i$ are $\boldsymbol{Ł}_n$-homomorphisms. The case of $[\,\alpha\,]\,\psi$ is established using the induction hypothesis (claim 2b) applied to α and (claim 1) to ψ, and the assumption that (1) holds for $[\,\alpha\,]\,\psi$.

The base case of claim 2a follows from the definitions of $I_{\mathfrak{N}}$ and I. The induction step is established easily using the induction hypothesis and the assumption that $N \subseteq M$. The case of test relies on Theorem 5.10.

The base case of claim 2b follows from the definition of I. The induction step follows easily from the assumption that $[\,\alpha\,]\,\psi$ satisfies (1) and the induction hypothesis. □

Lemma 6.2. *For each n and input $\chi \in \Phi_n$, Algorithm 1 terminates returning a $\boldsymbol{Ł}_n$-model isomorphic to $\mathfrak{M}_n^{FL(\chi)}$.*

Proof. The algorithm clearly terminates as M_0 is finite; part 2 of the algorithm applies a finite number of tests to each $h \in M_1$ and part 3 terminates after a finite number of iterations. Now let $\mathfrak{M}$ be the output of the algorithm. We prove that 1. $M = N$ and 2. $I_{\mathfrak{N}}(\alpha, h, g) = I(\alpha, h, g)$ for all $f, g \in N$. First, it is clear that $N \subseteq M_1$ and that no $h \in N$ gets erased from M in part 3 of the algorithm (this is established using Lemma 6.1, claim 2a); hence $N \subseteq M$ after part 3 terminates. Second, take any $g \in M$. Since $\mathfrak{M}$ is a model, the function $s : \Phi_n \to Ł_n$ defined by $s(\phi) := I(\varphi, h)$ is a n-correct valuation by Lemma 5.2, and so $s \in S_n$. It follows that $g = h(f_{|s|_\Gamma|}) \in N$. Hence, $M \subseteq N$. The second claim follows directly from the definition of $I_{\mathfrak{N}}$. As observed before, $\mathfrak{N}$ is (isomorphic to) $\mathfrak{M}_n^{FL(\chi)}$. □

Theorem 6.3. *There is a deterministic exponential-time algorithm that decides if a given $\chi \in \Phi_n$ is in* $\mathbf{PDŁ}_n$.

Proof. It follows from Theorems 5.10 and 5.11 that $\chi \in \mathbf{PDŁ}_n$ iff χ is valid in $\mathfrak{M}_n^{FL(\chi)}$. By Lemma 6.2, Algorithm 1 constructs a model $\mathfrak{N}$ isomorphic to $\mathfrak{M}_n^{FL(\chi)}$; the algorithm is easily seen to be terminating in time $O(n^{|FL'(\chi)|})$. Validity of χ in $\mathfrak{N}$ can be checked in polynomial time. (See (Fischer and Ladner 1979), Theorem 3.3 for the case $n = 2$; the general case can be established similarly.) □

6.2 A Lower Bound

The **PDL** validity problem is known to be *EXPTIME*-complete (Fischer and Ladner 1979). In this section we show that the validity problem for $\mathbf{PDŁ}_n$ is *EXPTIME*-hard for all $n > 1$; we give for each $n > 1$ a translation τ_n from $\mathcal{L}_2$, the classical language of **PDL**, to any $\mathcal{L}_n$ such that $\varphi \in \mathbf{PDL}$ iff $\tau_n(\varphi) \in \mathbf{PDŁ}_n$.

In what follows, an $\boldsymbol{Ł}_2$-frame (model) will be called also a *crisp* frame (model). Let $\mathfrak{M} = (S, E, V)$ be a $\boldsymbol{Ł}_n$-model for any $n \geq 2$. The *crisp variant* of $\mathfrak{M}$ is the crisp model $\mathfrak{M}^c = (S, E^c, V^c)$, where

- $E^c(i)(s,t) = 1$ if $E(i)(s,t) = 1$ and $= 0$ otherwise;
- $V^c(i)(s) = 1$ if $V(i)(s) = 1$ and $= 0$ otherwise.

Algorithm 1 An algorithm that returns, on input $\chi \in \Phi_n$, a model isomorphic to $\mathfrak{M}_n^{FL(\chi)}$.

Input: A formula $\chi \in \Phi_n$. **Return:** A $\boldsymbol{Ł}_n$-model $\mathfrak{M}$.

Part 1. Construct $FL(\chi)$; take $FL'(\chi)$, the set of $\varphi \in FL(\chi)$ such that either φ is a propositional variable, or φ is a formula of the form $[\,\alpha\,]\,\psi$; construct M_0, the set of all functions $f : FL'(\{\chi\}) \to Ł_n$;

(*Comment:* The cardinality of M_0 is $n^{|FL'(\chi)|}$. Each $f \in M_0$ extends uniquely to a $\boldsymbol{Ł}_n$-homomorphism $h(f)$ from $FL(\chi)$ to $Ł_n$ and each such homomorphism is an extension of some $f \in M_0$.)

Construct M_1, the set of all $\boldsymbol{Ł}_n$-homomorphisms from $FL(\chi)$ to $Ł_n$, by extending each $f \in M_0$ to a $\boldsymbol{Ł}_n$-homomorphism. End of Part 1.

Part 2. For all $h \in M_1$, check if h satisfies the following:

- if $[\,\alpha \cup \beta\,]\,\varphi \in \Gamma$, then $h([\,\alpha \cup \beta\,]\,\varphi) = \min\big(h([\,\alpha\,]\,\varphi), h([\,\beta\,]\,\varphi)\big)$;
- if $[\,\alpha;\beta\,]\,\varphi \in \Gamma$, then $h([\,\alpha;\beta\,]\,\varphi) = h([\,\alpha\,]\,[\,\beta\,]\,\varphi)$;
- if $[\,\alpha^*\,]\,\varphi \in \Gamma$, then $h([\,\alpha^*\,]\,\varphi) = \min\big(h(\varphi), h([\,\alpha\,]\,[\,\alpha^*\,]\,\varphi)\big)$;
- if $[\,\psi?\,]\,\varphi \in \Gamma$, then $h([\,\psi?\,]\,\varphi) = h(\psi \to \varphi)$.

Construct M_2, the set of all $h \in M_1$ that satisfy these requirements. End of Part 2.

Part 3. Let $\Delta = \{[\,\alpha\,]\,\varphi \in FL'(\chi) \mid \alpha \in A \text{ or } \exists\beta(\alpha = \beta^*)\}$; we assume that Δ is ordered by $Co(\alpha)$, increasingly.

Set $M := M_2$. Construct the $\boldsymbol{Ł}_n$ model $\mathfrak{M} = (M, E, I)$ where, for all $h, g \in M$,

- $E(j)(h,g) = \bigwedge^{\boldsymbol{Ł}_n}\{h([\,a_j\,]\,\varphi) \to^{\boldsymbol{Ł}_n} g(\varphi) \mid [\,a_j\,]\,\varphi \in \Gamma\}$;
- $V(j)(h) = h(p_j)$ if $p_j \in \Gamma$ and $= 0$ otherwise;
- I is defined as usual.

Assume some ordering of M. Search for the first $h \in M$ such that there is $[\,\alpha\,]\,\varphi \in \Delta$ such that

$$\bigwedge_{g \in M}^{\boldsymbol{Ł}_n} \{I(\alpha, h, g) \to^{\boldsymbol{Ł}_n} g(\varphi)\} \sqsubseteq h([\,\alpha\,]\,\varphi) \qquad (1)$$

is not satisfied. Return $\mathfrak{M}$ if no such h is found. Otherwise let h be the first element of M found in the search; assign $M := M \setminus \{h\}$, re-compute E, V and I, and repeat the search.

(*Comment:* It can be easily checked that tests $\psi?$ satisfy (1) for all h and φ and if α, β satisfy (1) for all h and φ, then so do $\alpha \cup \beta$ and $\alpha;\beta$. Hence we may work with Δ instead of the set of all modal formulas in $FL'(\chi)$.)

$I_{\mathfrak{M}^c}$ is defined as usual. It is clear that $\mathfrak{M}^c$ is a crisp model for each $\mathfrak{M}$.

For any $n \geq 2$, let τ_n be the function from $\mathcal{L}_2$-expressions to $\mathcal{L}_n$-expressions defined as follows ($\star \in \{\wedge, \vee, \to\}$ and

$\dagger \in \{\cup, ;\}$):

$$\tau_n(p) = p^n \qquad \tau_n(a) = a$$
$$\tau_n(c) = c \qquad \tau_n(\alpha \dagger \beta) = \tau_n(\alpha) \dagger \tau_n(\beta)$$
$$\tau_n(\varphi \star \psi) = \tau_n(\varphi) \star \tau_n(\psi) \qquad \tau_n(\alpha^*) = \tau_n(\alpha)^*$$
$$\tau_n([\alpha]\varphi) = n\big([\tau_n(\alpha)]\tau_n(\varphi)\big) \qquad \tau_n(\varphi?) = \tau_n(\varphi)?$$

Lemma 6.4. *Take any $\mathbf{Ł}_n$-model $\mathfrak{M}$. For each $\varphi \in \Phi_2$, $\alpha \in \Pi_2$ and s, t in $\mathfrak{M}$:*

1. *$I_{\mathfrak{M}^c}(\varphi, s) = 1$ iff $I_{\mathfrak{M}}(\tau_n(\varphi), s) = 1$;*
2. *$I_{\mathfrak{M}^c}(\alpha, s, t) = 1$ iff $I_{\mathfrak{M}}(\tau(\alpha), s, t) = 1$.*

Proof. Simultaneous induction on the complexity of expressions. We write I^c instead of $I_{\mathfrak{M}^c}$, I instead of $I_{\mathfrak{M}}$ and τ instead of τ_n.

The base cases follow from the definition of $\mathfrak{M}^c$. In the induction step for the first claim, we make use of the obvious fact that $I(\tau(\chi), s) \in \{0, 1\}$ for all $\chi \in \mathcal{L}_2$ (hence the extra n in the definition of $\tau([\alpha]\varphi)$). The claims for $c, \wedge$ and $\vee$ of the induction step are straightforward; moreover, $I^c(\varphi \to \psi, s) = 1$ iff $I^c(\varphi, s) \neq 1$ or $I^c(\psi, s) = 1$ iff $I(\tau(\varphi), s) \neq 1$ or $I(\tau(\psi), s) = 1$ (by the induction hypothesis) iff $I(\tau(\varphi \to \psi), s) = 1$. Note that the last equivalence does not hold if it is not guaranteed that $I(\tau(\chi), s) \in \{0, 1\}$ for all $\chi \in \mathcal{L}_2$. The modal case is established as follows:

$$
\begin{aligned}
& I^c([\alpha]\varphi, s) \neq 1 \\
\iff & \exists t \in S \left(I^c(\alpha, s, t) = 1 \;\&\; I^c(\varphi, t) = 0\right) \\
\iff & \exists t \in S \left(I(\tau(\alpha), s, t) = 1 \;\&\; I(\tau(\varphi), t) \neq 1\right) \\
\iff & \exists t \in S \left(I(\tau(\alpha), s, t) = 1 \;\&\; I(\tau(\varphi), t) = 0\right) \\
\iff & \bigwedge_{t \in S}^{\mathbf{Ł}_n} \left\{ I(\tau(\alpha), s, t) \to^{\mathbf{Ł}_n} I(\tau(\varphi), t) \right\} = 0 \\
\iff & I([\tau(\alpha)]\tau(\varphi), s) = 0 \iff I(n([\tau(\alpha)]\tau(\varphi)), s) = 0 \\
\iff & I(\tau([\alpha]\varphi), s) \neq 1
\end{aligned}
$$

The second equivalence uses the induction hypothesis (for both claims), the third one uses the fact that $I(\tau(\varphi), t) \in \{0, 1\}$ and the sixth one uses Lemma 2.3.

The cases for $\cup$, ; and $*$ in the induction step for the second claim are straightforward; moreover, $I^c([\psi?]\varphi) = 1$ iff $s = t$ and $I^c(\psi, t) = 1$ iff $s = t$ and $I(\tau(\psi), t) = 1$ iff $I(\tau(\psi)?, s, t) = 1$. □

Lemma 6.4 implies that $I_{\mathfrak{M}}(\varphi, s) = I_{\mathfrak{M}^c}(\varphi, s)$ if φ is a $\mathcal{L}_2$-formula and $\mathfrak{M}$ is a $\mathbf{Ł}_2$ model, since then $\mathfrak{M}^c = \mathfrak{M}$.

Lemma 6.5. *For all $\varphi \in \mathcal{L}_2$, $\varphi \in$ **PDL** iff $\tau_n(\varphi) \in$ **PDŁ**$_n$.*

Proof. If $\varphi \notin$ **PDL**, then there is a $\mathbf{Ł}_2$-model $\mathfrak{C}$ such that $I_{\mathfrak{C}}(\neg\varphi, s) = 1$ for some s and so $I_{\mathfrak{C}}(\tau_n(\neg\varphi), s) = 1$ for all n by Lemma 6.4. Hence, for all n there is a $\mathbf{Ł}_n$-model $\mathfrak{M}$ (namely, $\mathfrak{C}$) such that $I_{\mathfrak{M}}(\neg\tau_n(\varphi), s) = 1$. Hence, by Theorem 5.11, $\tau_n(\varphi) \notin$ **PDŁ**$_n$. Conversely, if $\tau_n(\varphi) \notin$ **PDŁ**$_n$, then there is a $\mathbf{Ł}_n$-model $\mathfrak{M}$ such that $I_{\mathfrak{M}}(\tau_n(\varphi), s) < 1$ for some s in $\mathfrak{M}$ by Theorem 5.11. Hence, $I_{\mathfrak{M}}(\tau_n(\neg\varphi), s) > 0$, which means that $I_{\mathfrak{M}}(\tau_n(\neg\varphi), s) = 1$. It follows by Lemma 6.4 that $I_{\mathfrak{M}^c}(\neg\varphi, s) = 1$ and so $\varphi \notin$ **PDL**. □

Theorem 6.6. *For each n, the problem whether $\varphi \in$ **PDŁ**$_n$ is EXPTIME-hard.*

Proof. Lemma 6.5 and the known fact that the satisfiability (and so the validity) problem for **PDL** is *EXPTIME*-hard. □

7 Conclusion

We studied many-valued propositional dynamic logics based on relational models where both satisfaction of formulas in states and accessibility between states are evaluated in a finite Łukasiewicz chain. For each **PDŁ**$_n$ where $n > 1$, we provided a sound and weakly complete recursive axiomatization and established its decidability using filtration; we generalized Pratt's exponential-time algorithm for checking validity in **PDL** (= **PDŁ**$_2$) to arbitrary $n > 1$ and we have shown that the validity problem for each **PDŁ**$_n$ is *EXPTIME*-hard by finding embeddings from **PDL** into **PDŁ**$_n$. This work extends the sparse existing results on many-valued dynamic logics and, we believe, lays the groundwork for future developments and applications.

Many interesting problems remain open; let us mention just two here. Firstly, it would be interesting to extend our results to many-valued dynamic logic based on infinitely-valued Łukasiewicz logic. An inspection of our completeness proof reveals that we have relied on finiteness of $\mathbf{Ł}_n$ and so our technique does not seem to be applicable to logics using $\mathbf{Ł}$, the uncountable Łukasiewicz chain based on the real interval $[0, 1]$. Secondly, in light of the suggestion of (Hájek 1998; Hájek, Godo, and Esteva 1995; Hájek, Godo, and Esteva 2000) to use many-valued modal logic to formalize reasoning about probabilities, it is interesting to explore versions of many-valued dynamic logic formalizing reasoning about probabilistic actions.[6] A first step toward this goal would be an exploration of dynamic logics extending combinations of infinitely-valued Łukasiewicz logic (or Rational Pavelka logic) with Product logic, and their two-layered modal extensions developed to formalize reasoning about probabilities in (Hájek 1998; Hájek, Godo, and Esteva 1995; Hájek, Godo, and Esteva 2000).

Acknowledgements

This work was supported by the Czech Science Foundation grant GJ18-19162Y for the project *Non-Classical Logical Models of Information Dynamics*. The author is grateful to three anonymous reviewers for valuable suggestions.

References

Baader, F. 1991. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles.

[6] As explained in (Hájek, Godo, and Esteva 1995) for instance, truth degrees differ from probabilities; however, one can consider statements such as "p is probable" or "it is probable that a will terminate in a state satisfying p" as statements admitting truth degrees determined by the underlying probabilities—see also (Hughes, Esterline, and Kimiaghalam 2006) for a many-valued dynamic logic reflecting this interpretation.

In *Proc. 12th International Joint Conference on Artificial Intelligence (IJCAI'91)*, 446–451.

Bacchus, F.; Halpern, J. Y.; and Levesque, H. J. 1999. Reasoning about noisy sensors and effectors in the situation calculus. *Artificial Intelligence* 111(1-2):171–208.

Běhounek, L. 2008. Modeling costs of program runs in fuzzified propositional dynamic logic. In Hakl, F., ed., *Doktorandské dny '08*, 11–18. ICS AS CR and Matfyzpress.

Blackburn, P.; de Rijke, M.; and Venema, Y. 2001. *Modal Logic*. Cambridge University Press.

Bou, F.; Esteva, F.; Godo, L.; and Rodríguez, R. O. 2011. On the minimum many-valued modal logic over a finite residuated lattice. *Journal of Logic and Computation* 21(5):739–790.

Cintula, P.; Noguera, C.; and Smith, N. J. J. 2017. A logical framework for graded predicates. In Baltag, A.; Seligman, J.; and Yamada, T., eds., *Proc. 6th International Workshop on Logic, Rationality and Interaction (LORI'17)*, 3–16. Springer.

Conradie, W.; Morton, W.; and Robinson, C. 2017. Filtrations for many-valued modal logic with applications. Presentation at TACL 2017, Prague.

Di Nola, A.; Grigolia, R.; and Vitale, G. 2020. Dynamic Łukasiewicz logic and dynamic MV-algebras. *International Journal of Approximate Reasoning* 124:103–110.

Fagin, R.; Halpern, J. Y.; Moses, Y.; and Vardi, M. Y. 1995. *Reasoning About Knowledge*. MIT Press.

Fischer, M. J., and Ladner, R. E. 1979. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences* 18:194–211.

Grigolia, R. 1977. Algebraic analysis of Łukasiewicz-Tarski's n-valued logical systems. In Wójcicki, R., and Malinowski, G., eds., *Selected Papers on Łukasiewicz Sentential Calcuuli*. Polish Academy of Sciences. 81–92.

Hájek, P.; Godo, L.; and Esteva, F. 1995. Fuzzy logic and probability. In *Proc. 11th Conference on Uncertainty in Artificial Intelligence (UAI'95)*, 237–244.

Hájek, P.; Godo, L.; and Esteva, F. 2000. Reasoning about probability using fuzzy logic. *Neural Network World* 10(5):811–824.

Harel, D.; Kozen, D.; and Tiuryn, J. 2000. *Dynamic Logic*. MIT Press.

Hughes, J.; Esterline, A.; and Kimiaghalam, B. 2006. Means-end relations and a measure of efficacy. *Journal of Logic, Language and Information* 15(1):83–108.

Hájek, P. 1998. *Metamathematics of Fuzzy Logic*. Kluwer.

Kozen, D., and Parikh, R. 1981. An elementary proof of the completeness of PDL. *Theoretical Computer Science* 14:113–118.

Kozen, D. 1985. A probabilistic PDL. *Journal of Computer and System Sciences* 30(2):162 – 178.

Laverny, N., and Lang, J. 2005a. From knowledge-based programs to graded belief-based programs, part II: Off-line reasoning. In *Proc. 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, IJCAI'05, 497–502. Morgan Kaufmann Publishers Inc.

Laverny, N., and Lang, J. 2005b. From knowledge-based programs to graded belief-based programs, part I: On-line reasoning. *Synthese* 147(2):277–321.

Liau, C.-J. 1999. Many-valued dynamic logic for qualitative decision theory. In Zhong, N.; Skowron, A.; and Ohsuga, S., eds., *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing*, 294–303. Springer Berlin Heidelberg.

Madeira, A.; Neves, R.; and Martins, M. A. 2016. An exercise on the generation of many-valued dynamic logics. *Journal of Logical and Algebraic Methods in Programming* 85(5, Part 2):1011–1037. Articles dedicated to Prof. J. N. Oliveira on the occasion of his 60th birthday.

McCarthy, J. 1963. Situations, actions and causal laws. Technical report, Stanford University.

Meyer, J.-J. C.; van der Hoek, W.; and van Linder, B. 1999. A logical approach to the dynamics of commitments. *Artificial Intelligence* 113(1-2):1–40.

Meyer, J.-J. C. 1987. A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic* 29(1):109–136.

Meyer, J.-J. C. 2000. Dynamic logic for reasoning about actions and agents. In Minker, J., ed., *Logic-Based Artificial Intelligence*. Springer US. 281–311.

Mio, M., and Simpson, A. 2017. Łukasiewicz μ-calculus. *Fundamenta Informaticae* 150(3-4):317–346.

Pratt, V. R. 1979. Models of program logics. In *Proc. 20th Annual Symposium on Foundations of Computer Science (FOCS'79)*, FOCS '79, 115–122. IEEE Computer Society.

Reiter, R. 2001. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press.

Rosenschein, S. J. 1981. Plan synthesis: A logical perspective. In *Proc. 7th International Joint Conference on Artificial Intelligence (IJCAI'81)*, 331–337.

Sedlár, I. 2020. Finitely-valued propositional dynamic logic. In Olivetti, N.; Verbrugge, R.; Negri, S.; and Sandu, G., eds., *Proc. 13th International Conference on Advances in Modal Logic (AiML'20)*, 561–579. College Publications.

Smith, N. J. J. 2008. *Vagueness and Degrees of Truth*. Oxford University Press.

Spalazzi, L., and Traverso, P. 2000. A dynamic logic for acting, sensing, and planning. *Journal of Logic and Computation* 10(6):787–821.

Teheux, B. 2014. Propositional dynamic logic for searching games with errors. *Journal of Applied Logic* 12(4):377–394.

van der Hoek, W., and Meyer, J. J. C. 1992. Graded modalities in epistemic logic. In Nerode, A., and Taitslin, M., eds., *Proc. 2nd International Symposium on Logical Foundations of Computer Science (LFCS'92)*, 503–514. Springer Berlin Heidelberg.

van der Hoek, W.; van Linder, B.; and Meyer, J.-J. C. 1994. A logic of capabilities. In *Proc. 3rd International Symposium on Logical Foundations of Computer Science (LFCS'94)*, 366–378. Springer.

Wooldridge, M. J. 2000. *Reasoning about Rational Agents*. The MIT Press.

Zhang, X.; Jiang, M.; Zhou, C.; and Hao, Y. 2012. Graded BDI models for agent architectures based on łukasiewicz logic and propositional dynamic logic. In Wang, F. L.; Lei, J.; Gong, Z.; and Luo, X., eds., *Proc. 4th International Conference on Web Information Systems and Mining (WISM'12)*, 439–450. Springer Berlin Heidelberg.