Non-Classical PDL on the Cheap

IGOR SEDLÁR[1]

Abstract: A four-valued version of PDL is obtained by defining a DeMorgan negation as a negative modality using a special atomic program. Simple examples indicating applications of the formalism in formal verification of epistemic programs and epistemic planning are provided. Decidability and weak completeness are established.

Keywords: DeMorgan negation, Epistemic planning, Formal verification, Propositional dynamic logic

1 Introduction

Propositional dynamic logic PDL is a well-known formalism for specifying and verifying properties of regular programs (Fischer & Ladner, 1979; Harel, Kozen, & Tiuryn, 2000). In the Kripke-style semantics for PDL, programs are represented by state transitions ('state y is a possible outcome of executing program α in state x') where states of the computer are taken to be complete and consistent possible worlds. Technically speaking, in PDL states correspond to functions from the set of formulas to the set of classical truth values $\{0, 1\}$: every formula is either false (0) or true (1), but not both.

Several generalisations of this approach have been suggested. In this article we shall focus on the one put forward by Belnap and Dunn (Belnap, 1977a, 1977b; Dunn, 1976). Belnap–Dunn states correspond to functions from formulas to *subsets* of $\{0, 1\}$; and are seen as *bodies of information about the world* rather than possible states of the world (possible worlds). On this view, the four possible truth values $\emptyset, \{0\}, \{1\}$ and $\{0, 1\}$ correspond to four possible answers to queries about a formula with respect to a fixed body of information: the body of information does not provide any information about the formula ($\emptyset$); it provides information that the formula is false and no information that it is true ($\{0\}$); it provides information that the formula is true and no information that it is false ($\{1\}$); it provides conflicting information about the formula ($\{0, 1\}$). The notion of a computer

[1]This work was supported by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807).

program operating on Belnap–Dunn states is natural since such programs can be seen as algorithmic transformations of database-like bodies of information, central to areas such as epistemic planning.[2]

Sedlár (2016) outlines a version of PDL over an extension of the Belnap–Dunn logic studied by Odintsov and Wansing (2010). This article introduces a simplification of the approach. It is shown that a version of PDL corresponding to the Belnap–Dunn notion of state can be defined by a very simple modification of standard PDL. Building on the approach of Fagin, Halpern, and Vardi (1995), we extend PDL with a *modal DeMorgan negation* connective '$\sim$', interpreted semantically by the Routley star operator (Routley & Routley, 1972). We construe the Routley star as a serial, symmetric and functional atomic program. Decidability and weak completeness of the resulting system are established.

We note that (Sedlár, 2016) and the present article can be seen as an addition to the small but growing literature on non-classical PDL. We should mention Teheux (2014) who formulates PDL over finitely-valued Łukasiewicz logics to model the Rényi–Ulam searching game with errors. Baltag and Smets (2006) present a dynamic logic for reasoning about information flow in quantum programs and Bergfeld and Sack (2015) discuss a probabilistic logic for quantum programs.

The article is organised as follows. Section 2 outlines the basics of standard PDL. Section 3 discusses a modal logic with De Morgan negation and Section 4 extends this logic to PDL with De Morgan negation, NPDL. Section 5 outlines potential applications of NPDL. Section 6 provides a brief summary of the article.

2 Standard PDL

This section outlines standard PDL using complete and consistent possible worlds as representations of the states programs operate on. For more details, see (Harel et al., 2000).

The language $\mathcal{L}$ consists of two classes of expressions, namely, programs ($\alpha \in P$) and formulas ($\phi \in F$), defined as follows:

$$\alpha ::= a \mid \alpha ; \alpha \mid \alpha \cup \alpha \mid \alpha^* \mid \phi? \qquad \phi ::= p \mid \neg\phi \mid \phi \to \phi \mid [\alpha]\phi$$

[2] To give another example of a generalisation of the classical notion of state, if some formulas correspond to statements involving imprecise or graded predicates, then functions from the set of formulas to the real interval $[0, 1]$ are the appropriate formalisation.

($a \in AP$, a countable set of atomic programs and $p \in AF$, a countable set of atomic formulas). Formulas of the form $[\alpha]\phi$ are read 'It is necessary that after executing α, ϕ will hold'. The operator ';' is seen as program composition ('Execute α, then execute β'), '$\cup$' as non-deterministic choice ('Choose either α or β nondeterministically and execute it'), '$*$' as iteration ('Execute α a nondeterministically chosen finite number of times') and '?' as test ('Test whether ϕ is the case; proceed if true, fail if false'). As usual, $\phi \wedge \psi \stackrel{def}{=} \neg(\phi \to \neg\psi)$, $\phi \vee \psi \stackrel{def}{=} \neg(\neg\phi \wedge \neg\psi)$ and $\langle\alpha\rangle\phi \stackrel{def}{=} \neg[\alpha]\neg\phi$. Models for this language ('*dynamic models*') are multi-dimensional Kripke models $M = \langle S, R, V\rangle$, where every α is assigned a binary relation $R(\alpha)$ on the set of states S and every ϕ is assigned a subset $V(\phi)$ of S as follows: $V(p)$ is arbitrary; $V(\neg\phi)$ is the complement of $V(\phi)$; $V(\phi \to \psi) = (S - V(\phi)) \cup V(\psi)$; $V([\alpha]\phi)$ is the set of $x \in S$ such that, for all y, if $\langle x, y\rangle \in R(\alpha)$, then $y \in V(\phi)$ ('the set of such states x that every successful (terminating, halting) execution of α in x results in a state that satisfies ϕ'); $R(\alpha;\beta)$ ($R(\alpha \cup \beta)$) is the composition (union) of $R(\alpha)$ and $R(\beta)$; $R(\alpha^*)$ is the reflexive transitive closure of $R(\alpha)$; and $R(\phi?)$ is the identity relation on $V(\phi)$. (Note that $x \in V(\langle\alpha\rangle\phi)$ iff there is y such that $R(\alpha)xy$ and $y \in V(\phi)$, i.e., $\langle\alpha\rangle\phi$ means that it is possible that after executing α, ϕ will hold.) V could have been defined equivalently as a function from S to functions from F to $\{0, 1\}$ (as discussed in the introduction), but we have chosen this simpler formulation. Sometimes the notation 'S_M', 'R_M' and 'V_M' is used to make the relevant M explicit.

The following metalogical notions are used, as defined here, throughout the article. A formula ϕ is valid in model M iff $V_M(\phi) = S_M$ (notation: $M \models \phi$). A formula ϕ follows from (is entailed by) Γ as the set of local assumptions and Δ as the set of global assumptions (notation: $[\Delta], \Gamma \models \phi$) iff

$$\bigcap_{\psi \in \Gamma} V_M(\psi) \subseteq V_M(\phi)$$

for all M such that $M \models \Delta$ (i.e., $M \models \chi$ for all $\chi \in \Delta$). A formula ϕ follows from Γ globally (locally) iff $[\Gamma], \emptyset \models \phi$ ($[\emptyset], \Gamma \models \phi$). We also use the notation $[\Gamma] \models \phi$ ($\Gamma \models \phi$) for global (local) entailment. A formula ϕ is valid in a class of models $\mathcal{C}$ iff $M \models \phi$ for all $M \in \mathcal{C}$; ϕ is valid in PDL iff it is valid in the class of all dynamic models (notation: $\models \phi$). A formula ϕ is satisfiable iff there is $\mathcal{M}$ such that $M \not\models \neg\phi$.

The language $\mathcal{L}$ is able to express a number of standard programming constructs. For example, *if* ϕ *then* α *else* β translates to $(\phi?;\alpha) \cup (\neg\phi?;\beta)$

and *while* ϕ *do* α to $(\phi?;\alpha)^*;\neg\phi?$. *Correctness* of programs (given some desired functionality) can be seen as a relation between specific inputs and outputs of terminating executions of the program: given an input specified by a formula ϕ, every terminating (successful) execution of the program terminates in a state satisfying ψ. In this case, we say that a program α is *partially correct* with respect to precondition ϕ and postcondition ψ. Partial correctness assertions are expressible in $\mathcal{L}$ as formulas of the form

$$\phi \to [\alpha]\psi \tag{1}$$

Now PDL can be used to check whether specific partial correctness assertions follow from some given global assumptions Δ (thought of as 'invariant' assumptions holding in the initial state and every possible outcome of every possible computation) and some local assumptions Γ (required to hold only in the initial state of the computation): $[\Delta],\Gamma \models^? \phi \to [\alpha]\psi$. The key observation is that, in most practically interesting cases, such questions are decidable.

Proposition 1 *Let* $\Gamma,\Delta \subseteq F_{\mathcal{L}}$ *be finite and* $\phi \in F_{\mathcal{L}}$. *If* $\{a_1,\ldots,a_n\}$ *is the set of all atomic programs appearing in some formula in* $\Delta\cup\Gamma$ *or in* ϕ, *then*

$$[\Delta],\Gamma \models \phi \quad\iff\quad \models \left[(a_1\cup\ldots\cup a_n)^*\right]\bigwedge\Delta \to \left(\bigwedge\Gamma \to \phi\right)$$

Proof. Let us write the claim on the right-hand side as $\models \Delta^* \to (\bigwedge\Gamma \to \phi)$. The right-to-left implication is trivial. If $M \models \Delta$ and $x \in V_M(\bigwedge\Gamma)$, then obviously $x \in V_M(\Delta^* \wedge \bigwedge\Gamma)$. So $x \in V_M(\phi)$ follows from the assumption.

To establish the converse implication, assume that $\Delta^* \to (\bigwedge\Gamma \to \phi)$ is not valid. This means that there are some M and $x \in S_M$ such that $x \in V_M(\Delta^* \wedge \bigwedge\Gamma)$ and $x \notin V_M(\phi)$. Now define $M_x = \langle S_x, R_x, V_x\rangle$ as follows: $S_x = \{y \mid \langle x,y\rangle \in R((a_1\cup\ldots\cup a_n)^*)\}$; $R_x(\alpha) = R(\alpha)\cap S_x^2$ for all α; $V_x(p) = V(p)\cap S_x$ and $V_x(\phi)$ for complex ϕ are build up recursively as in the definition of dynamic model. It is plain that $M_x \models \Delta$, but $M_x \not\models \bigwedge\Gamma \to \phi$. (The key fact, easily established by induction on the complexity of α, is that if every atomic program appearing in α is in $\{a_1\cup\ldots\cup a_n\}$, then $R(\alpha)zz'$ only if $R_x(\alpha)zz'$, for all $z,z' \in S_x$.) $\square$

Theorem 1 *The validity problem for* PDL *(*$\models^? \phi$*) is decidable (EXPTIME-complete).*

Proof. See (Harel et al., 2000), chapters 6 and 8. $\square$

Hence, for any finite sets of global assumptions Δ, local assumptions Γ and any partial correctness assertion $\phi \to [\alpha]\psi$, there is an algorithm running in time exponential to the size of the input and deciding whether the assertion follows from the assumptions. For infinite Δ, the situation gets worse; it is even sufficient to consider the set $SI(\phi)$ of substitution instances of some fixed ϕ.

Theorem 2 *The question whether* $[SI(\phi)], \emptyset \models \psi$ *is undecidable (*Π^1_1*-complete).*

Proof. See (Harel et al., 2000), chapter 8.3. □

Because of the 'infinitary' iteration operator $*$, PDL is not compact.[3] Hence, there is no hope to provide a strongly complete axiomatization. However, weak completeness is another story.

Theorem 3 *Let* $H(\mathsf{PDL})$ *be the Hilbert-style system extending any axiomatisation of classical propositional logic in* $\{\neg, \to\}$ *by schemas*

1. $[\alpha](\phi \to \psi) \to ([\alpha]\phi \to [\alpha]\psi)$
2. $[\alpha \cup \beta]\phi \leftrightarrow ([\alpha]\phi \wedge [\beta]\phi)$
3. $[\alpha; \beta]\phi \leftrightarrow [\alpha][\beta]\phi$
4. $[\psi?]\phi \leftrightarrow (\psi \to \phi)$
5. $[\alpha^*]\phi \leftrightarrow (\phi \wedge [\alpha][\alpha^*]\phi)$
6. $(\phi \wedge [\alpha^*](\phi \to [\alpha]\phi)) \to [\alpha^*]\phi$

and the Necessitation rule $\phi/[\alpha]\phi$*. Let theoremhood in* $H(\mathsf{PDL})$ *be defined as usual. Then* ϕ *is a theorem of* $H(\mathsf{PDL})$ *iff* $\models \phi$*.*

Proof. See (Harel et al., 2000), chapter 7. □

[3] Every finite subset of $\{\langle\alpha^*\rangle\phi\} \cup \{\neg\phi, \langle\alpha\rangle\phi, \langle\alpha;\alpha\rangle\phi, \ldots\}$ is satisfiable but the set itself is not satisfiable.

3 Modal logic with De Morgan negation

This section outlines the framework of Fagin et al. (1995). The language $\mathcal{L}_n^{\sim}$ is a fragment of $\mathcal{L}$ without program operators ';', '$\cup$', '$*$' and '?', with only a finite $AP_n \subseteq AP$ of cardinality n, extended by a new unary connective '$\sim$'. (Informally, members of AP_n are seen as agents, but this is not important for our purposes.) *Star models* are $\mathcal{M} = \langle S, R, \star, V \rangle$, where S and R are as before (the range of R is AP_n), $\star$ is a unary operation of period two (i.e., $(x^\star)^\star = x$) on S and V is defined as before with the addition of

$$V(\sim\phi) = \{x \mid x^\star \notin V(\phi)\} \tag{2}$$

Let us denote the set of $\mathcal{L}_n^{\sim}$-formulas valid in every star model as NK. It is plain that '$\sim$' does not satisfy many of laws adhered by classical negation '$\neg$'. For instance, not every formula of the form $\sim\phi \vee \phi$ or $(\phi \wedge \sim\phi) \to \psi$ is in NK, but $\sim\sim\phi \leftrightarrow \phi$ and all of the De Morgan laws are in NK.[4] Hence, '$\sim$' turns out to be what is usually called a De Morgan negation (Dunn, 1993).

Theorem 4 *Let $H(\mathsf{NK})$ be a Hilbert-style system that extends any axiomatization of classical propositional logic in $\{\neg, \to\}$ by schemas*

1. $[\alpha](\phi \to \psi) \to ([\alpha]\phi \to [\alpha]\psi)$
2. $\sim\sim\phi \leftrightarrow \phi$
3. $\sim(\phi \wedge \psi) \leftrightarrow (\sim\phi \vee \sim\psi)$
4. $\sim(\phi \vee \psi) \leftrightarrow (\sim\phi \wedge \sim\psi)$

and the Necessitation rule $\phi/[\alpha]\phi$. Then ϕ is a theorem of $H(\mathsf{NK})$ iff $\mathcal{M} \models \phi$ for every star model $\mathcal{M}$.

Proof. See (Fagin et al., 1995). It the paper, a slightly different but equivalent axiomatization is used. □

Theorem 5 *Membership in* NK *(the problem of validity in every star model) is decidable (PSPACE-complete).*

Proof. See (Fagin et al., 1995). □

[4]Standard Kripke models can be seen as a special case where $x^\star = x$.

Note that this framework is, in effect, a fragment of PDL. Syntactically, the set of programs is limited to $AP_{n+1} = \{a_1, \ldots, a_n, b\}$ and $\sim\phi$ defined as $[b]\neg\phi$. Semantically, $R(b)$ is required to be a total function of period two, i.e., a relation that is *serial* $((\forall x)(\exists y)(Rxy))$, *symmetric* $((\forall xy)(Rxy \Rightarrow Ryx))$ and *functional* $((\forall xyz)((Rxy \;\&\; Rxz) \Rightarrow y = z))$.

4 PDL with De Morgan negation

Now consider the full language $\mathcal{L}$ again. Let us fix an atomic program $a \in AP$ and denote it as '$\star$'. Define $\sim\phi$ as $[\star]\neg\phi$ ('It is necessary that after executing $\star$, ϕ will not hold'). A *Routley model* is a dynamic model where $R(\star)$ is a total function of period two (serial, symmetric and functional). We reiterate that, when considering Routley models, we can meaningfully say $y = x^\star$ instead of $R(\star)xy$. Formula ϕ is said to be valid in NPDL iff it is valid in the class of Routley models.

Proposition 2 *Any formula of one of the following forms is valid in* NPDL*:*

1. $\langle\star\rangle\phi \leftrightarrow [\star]\phi$
2. $\phi \to [\star]\langle\star\rangle\phi$

Proof. If $R(\star)$ is functional, then, for every x, there is at most one y such that $R(\star)xy$. So if $x \in V(\langle\star\rangle\phi)$, then $y \in V(\phi)$ for every y such that $R(\star)xy$. Hence, $\langle\star\rangle\phi \to [\star]\phi$ is valid in NPDL. The converse $[\star]\phi \to \langle\star\rangle\phi$ is valid since $R(\star)$ is serial and $\phi \to [\star]\langle\star\rangle\phi$ is valid because $R(\star)$ is symmetric (as elementary modal logic has it). □

Corollary 1 *Any formula of one of the following forms is valid in* NPDL*:*

1. $\phi \to [\star][\star]\phi$
2. $[\star]\langle\star\rangle\phi \to \phi$

Proof. The first fact is obvious. To see that the second one holds as well, observe that

$$\begin{aligned}
(\forall\phi)(M \models \phi \to [\star]\langle\star\rangle\phi) &\implies (\forall\phi)(M \models \neg\phi \to [\star]\langle\star\rangle\neg\phi)\\
&\implies (\forall\phi)(M \models \neg[\star]\langle\star\rangle\neg\phi \to \neg\neg\phi)\\
&\implies (\forall\phi)(M \models \langle\star\rangle[\star]\phi \to \phi)\\
&\implies (\forall\phi)(M \models [\star][\star]\phi \to \phi)\\
&\implies (\forall\phi)(M \models [\star]\langle\star\rangle\phi \to \phi)
\end{aligned}$$

□

Corollary 2 *Any formula of one of the following forms is valid in* NPDL*:*

1. $\sim\sim\phi \leftrightarrow \phi$
2. $\sim(\phi \wedge \psi) \leftrightarrow (\sim\phi \vee \sim\psi)$
3. $\sim(\phi \vee \psi) \leftrightarrow (\sim\phi \wedge \sim\psi)$

Proof. (a) boils down to $[\star]\langle\star\rangle\phi \leftrightarrow \phi$; (b) is equivalent to $\langle\star\rangle(\neg\phi \vee \neg\psi) \leftrightarrow (\langle\star\rangle\neg\phi \vee \langle\star\rangle\neg\psi)$; and (c) is equivalent to $[\star](\neg\phi \wedge \neg\psi) \leftrightarrow ([\star]\neg\phi \wedge [\star]\neg\psi)$. □

If $\sim\phi$ is read as 'ϕ is false', then states in Routley models can be seen as Belnap–Dunn states. To be more specific, there are four possibilities for every x:

- $x \in V(\phi)$ and $x^\star \in V(\phi)$, so $x \in V(\phi \wedge \neg\sim\phi)$;
- $x \notin V(\phi)$ and $x^\star \notin V(\phi)$, so $x \in V(\neg\phi \wedge \sim\phi)$;
- $x \in V(\phi)$ and $x^\star \notin V(\phi)$, so $x \in V(\phi \wedge \sim\phi)$;
- $x \notin V(\phi)$ and $x^\star \in V(\phi)$, so $x \in V(\neg\phi \wedge \neg\sim\phi)$.

For every Routley model M and $x \in S_M$, let f_x^M be a function from formulas to subsets of $\{0, 1\}$ defined by setting $1 \in f_x^M(\phi)$ iff $x \in V(\phi)$ and $0 \in f_x^M(\phi)$ iff $x \in V(\sim\phi)$. The four possibilities specified above correspond to:

- $f_x^M(\phi) = \{1\}$,
- $f_x^M(\phi) = \{0\}$,
- $f_x^M(\phi) = \{0, 1\}$ and
- $f_x^M(\phi) = \emptyset$,

respectively. (Note that defining $\sim_1, \ldots, \sim_n$ in terms of $\star_1, \ldots, \star_n$ would enable to simulate 2^n-valued states for every finite n.[5])

Informally, the present framework treats the Routley star operator (Routley & Routley, 1972) as a program operating on database-like bodies of

[5] Thanks to Kit Fine for a suggestion along these lines.

information.[6] For the sake of simplicity, let us represent these bodies of information by functions from formulas to subsets of $\{0, 1\}$.[7] For every such f, define $f^\star$ by

$$f^\star(\phi) = \{\xi \in \{0,1\} \mid f(\phi) \cup \{\xi\} \neq \{0,1\}\}$$

In other words, $f^\star$ assigns to ϕ those 'classical' truth values that can be consistently added to the 'classical' truth values assigned to ϕ by f. Observe that $0 \in f(\phi)$ iff $1 \notin f^\star(\phi)$. In general,

$$\{0\} \xrightarrow{f^\star} \{0\} \qquad \{1\} \xrightarrow{f^\star} \{1\} \qquad \{0,1\} \xrightarrow{f^\star} \emptyset \qquad \emptyset \xrightarrow{f^\star} \{0,1\}$$

Hence, in a sense, the Routley star program corresponds to what has been called *conflation* in the literature on bilattices (Arieli & Avron, 1996).

Routley models comply with this interpretation, as witnessed by the following fact.

Proposition 3 *Any formula of one of the following forms is valid in* NPDL*:*

1. $(\phi \wedge [\star]\phi) \to [\star](\phi \wedge [\star]\phi)$
2. $(\phi \wedge [\star]\neg\phi) \to [\star](\neg\phi \wedge [\star]\phi)$
3. $(\neg\phi \wedge [\star]\phi) \to [\star](\phi \wedge [\star]\neg\phi)$
4. $(\neg\phi \wedge [\star]\neg\phi) \to [\star](\neg\phi \wedge [\star]\neg\phi)$

Proof. For all ϕ, ψ, $\phi \wedge [\star]\psi$ (both locally and globally) entails $[\star][\star]\phi \wedge [\star]\psi$ by Corollary 1, which entails $[\star](\psi \wedge [\star]\phi)$ by elementary modal logic. $\square$

Theorem 6 *The* NPDL *validity problem is decidable.*

Theorem 7 *Let* $H(\mathsf{NPDL})$ *be* $H(\mathsf{PDL})$ *extended by the schemas* $\langle\star\rangle\phi \leftrightarrow [\star]\phi$ *and* $\phi \to [\star]\langle\star\rangle\phi$*. Then* ϕ *is valid in* NPDL *iff it is a theorem of* $H(\mathsf{NPDL})$.

[6] Some find the Routley star in need of a convincing informal interpretation. For example, Dunn (1976) expresses worries whether the Routley star can be seen as something more that just a purely technical device without a reasonable informal interpretation.

[7] This is a simplifying assumption as, of course, there are Routley models M that contain a pair of states $x \neq y$ such that $f_x^M = f_y^M$. We can say that sates are characterised by these functions up to $\mathcal{L}$-equivalence, where x and y are $\mathcal{L}$-equivalent iff $x \in V(\phi) \iff y \in V(\phi)$ for all $\phi \in F_{\mathcal{L}}$.

The proofs of both of these theorems add little technical novelty to the standard proofs for PDL. The only thing that requires modification is the definition of the set used in filtration of the canonical model. In the proof for PDL, the so-called Fisher–Ladner closure $FL(\phi)$ of a given non-provable ϕ is used. However, it is easy to show that while $R(\star)$ in the canonical model is functional, $R(\star)^{FL(\phi)}$ in the filtrated model may fail to be a function. The solution to this problem lies in using a special finite superset $FLR(\phi)$ of $FL(\phi)$, also called the Fisher–Ladner–Routley closure of ϕ. The interested reader is referred to Appendix A.

5 Some examples

This section outlines some simple examples that demonstrate the expressivity (and possible applications) of NPDL. In these examples we use 'ϕ is supported (by x)' and 'There is information (in x) that ϕ is true' interchangeably. Instead of calling x a database-like body of information, we refer to it as a database. We say that ϕ is *decided* (by x) if ϕ or $\sim\phi$ is supported (by x) and that ϕ is *undecided* (by x) if ϕ is not decided (by x). We write ϕ^+ instead of $\phi \vee \sim\phi$ and ϕ^- instead of $\neg(\phi^+)$.

Being a 'special case' of PDL, NPDL can be used to check correctness of programs. In the context of NPDL, however, programs are seen as operating on bodies of information about the world, not the world itself. We may call these programs *epistemic programs*.[8]

Example 1 If states are seen as databases, then perhaps the simplest kinds of epistemic programs correspond to *adding* and *removing* ϕ from the database. Other simple examples are *testing* whether the database supports or decides a specific ϕ. The test programs are representable as $\phi?$ and $(\phi^+)?$, respectively. Representing addition and removal of information is trickier, but it can be done as follows. We may assume that, for a specific fixed ϕ, two (distinct) atomic programs a, b correspond to adding and removing ϕ, respectively. This assumption can be formalised by taking the formulas $\neg\phi \rightarrow [a]\phi$ and $\phi \rightarrow [b]\neg\phi$ as global assumptions. The role of global assumptions is perhaps best understood in contrast to the test programs. For instance, if we want to check whether $\phi?$ is partially correct with respect to precondition ψ and postcondition $\phi \wedge \psi$, then we ask whether $\psi \rightarrow [\phi?](\phi \wedge \psi)$ is valid in NPDL (the answer is, obviously, 'yes'). Now assume that we

[8]This terminology is related to but somewhat different in meaning from the one used by Baltag and Moss (2004).

want to check whether $a; b$ is partially correct with respect to $\neg\phi$ as both the precondition and postcondition. Obviously, $\neg\phi \to [a; b]\neg\phi$ is not valid in NPDL, but this is not the intuitively correct answer. However,

$$[\neg\phi \to [a]\phi, \phi \to [b]\neg\phi] \models \neg\phi \to [a; b]\neg\phi$$

holds.[9]

Example 2 An interesting special case of a postcondition is ϕ^+. Formulas of the form $\psi \to [\alpha]\phi^+$ say that every terminating execution of α where the input is a ψ-supporting database is a database that decides ϕ. Note that, in a sense, partial correctness claims play an important role in 'strategic' assessments of programs. For example, if the *goal* is to have a database that decides ϕ and

$$[\Gamma] \models \psi \to [\alpha]\phi^+ \quad \text{but} \quad [\Gamma] \not\models \psi \to [\beta]\phi^+,$$

(where Γ is some set of relevant global assumptions) then running α on a ψ-database is a *better choice* than running β. This is related to *epistemic planning*, i. e., the activity of choosing the appropriate course of actions given specific epistemic goals. Thus, NPDL can be seen as a formalism for checking the correctness of epistemic plans.

Epistemic planning, a 'special case' of automated planning[10] (Ghallab et al., 2004), has recently been formalised using the framework of dynamic epistemic logic (Bolander & Andersen, 2011). However, this approach has the disadvantage that checking the correctness of epistemic plans is undecidable even in the single-agent case if 'knowledge' is modelled by an epistemic logic weaker that S5 (Aucher & Bolander, 2013). Another advantage of NPDL as a formalism for epistemic planning is that it is able to distinguish between *different 'kinds' of inconsistency*. For instance, assume that p represents some irrelevant piece of information and q represents some rather important one. It is natural to assume that the presence of conflicting

[9]If $x \in V_M(\neg\phi)$ and $M \models \neg\phi \to [a]\phi$, then $R(a)xy$ only if $y \in V_M(\phi)$. But then, if also $M \models \phi \to [b]\neg\phi$, then $R(b)yz$ only if $z \in V_M(\neg\phi)$. Therefore, if $x \in V_M(\neg\phi)$, then $R(a; b)xz$ only if $z \in V_M(\neg\phi)$. The argument also illustrates why it is not sufficient to take $\neg\phi \to [a]\phi$ and $\phi \to [b]\neg\phi$ as local assumptions. If they were local, we could not apply $\phi \to [b]\neg\phi$ to y.

[10]We note that the 'plans as programs' approach, related to the present discussion, is the prevalent approach in deductive planning and there are applications of PDL in the area, see (Ghallab, Nau, & Traverso, 2004, Ch. 12) and (Rosenschein, 1981; Stephan & Biundo, 1993).

information about p in a database requires a different kind of action (probably none) than inconsistency concerning q. Planning formalisms based on normal modal logic (such as dynamic epistemic logic) recognize only one inconsistent database – the empty set. Hence, epistemic planning based on these approaches cannot diversify plans according to the 'seriousness' of the inconsistency involved.

6 Conclusion

This article explored the possibility to adapt PDL to modelling programs that operate on possibly incomplete and inconsistent bodies of information. It turns out that there is quite a simple way to do this, namely, defining a De Morgan negation in terms of a special atomic program $\star$, seen as the Routley star operation. The only extra assumptions needed pertain to $R(\star)$, which is assumed to be serial, symmetric and functional (function of period two). The resulting extension of PDL, NPDL, is decidable and has a sound and weakly complete axiomatisation (any axiomatisation of PDL plus the obvious axioms defining symmetry, seriality and functionality of $R(\star)$), as shown by simple modifications of the standard proofs for PDL. Last, but not least, simple examples of expressivity indicate that NPDL (and similar formalisms, e.g., the Belnapian PDL of Sedlár (2016)) might find applications in epistemic planning and related areas.

A Decidability and completeness of NPDL

This appendix contains (the interesting parts of) proofs of Theorems 6 and 7. The proofs are very close to similar proofs for standard PDL, the only difference being the assumptions concerning $R(\star)$. Of course, these assumptions need to be complied with when constructing the canonical model and defining filtrations.

We start by proving Theorem 6.

Definition 1 *Let $FL(\phi)$ be the Fisher-Ladner closure of ϕ (Harel et al., 2000, ch. 6.1). The* Fisher-Ladner-Routley closure *of ϕ, $FLR(\phi)$, is defined as*

$$FLR(\phi) = FL(\phi) \cup \{[\star]\psi \mid \psi \in FL(\phi) \textit{ and } \psi \neq [\star]\chi, \textit{for all } \chi\}$$

Hence, $FLR(\phi)$ is $FL(\phi)$ extended by $[\star]\psi$ for every $\psi \in FL(\phi)$ such that ψ does not begin with '$[\star]$'. It is plain that $FLR(\phi)$ is finite for all ϕ

(as $FL(\phi)$ is). Our reason for working with FLR instead of FL is that we want the filtrated $R(\star)$ to be a function of period two, but this is not the case if the canonical model is filtered trough $FL(\phi)$. We show later (Lemma 4) that the problem does not arise when FL is replaced by FLR.

Our main task now is to check that FLR has all the important properties of FL. In fact, the original proofs need to be modified only slightly. To be closer to the original proofs, we assume in this appendix that, as in (Harel et al., 2000), '$\bot$' and '$\to$' are primitive and the rest of the Boolean connectives are defined using these two in the usual manner.

Lemma 1 *If $\psi \to \chi \in FLR(\phi)$, then $\psi \in FLR(\phi)$ and $\chi \in FLR(\phi)$.*

Proof. If $\psi \to \chi \in FLR(\phi)$, $\psi \to \chi \in FL(\phi)$ by the definition of FLR. But then, by Lemma 6.1. of (Harel et al., 2000, ch. 6.1), $\{\psi, \chi\} \subseteq FL(\phi) \subseteq FLR(\phi)$. □

Lemma 2

1. *If $[\alpha]\psi \in FLR(\phi)$, then $\psi \in FLR(\phi)$*
2. *If $[\chi?]\psi \in FLR(\phi)$, then $\chi \in FLR(\phi)$*
3. *If $[\alpha \cup \beta]\psi \in FLR(\phi)$, then $[\alpha]\psi \in FLR(\phi)$ and $[\beta]\psi \in FLR(\phi)$*
4. *If $[\alpha;\beta]\psi \in FLR(\phi)$, then $[\alpha][\beta]\psi \in FLR(\phi)$ and $[\beta]\psi \in FLR(\phi)$*
5. *If $[\alpha^*]\psi \in FLR(\phi)$, then $[\alpha][\alpha^*]\psi \in FLR(\phi)$*

Proof. The only case to check is $[\star]\psi \in (FLR(\phi) \setminus FL(\phi))$. However, this case arises only if $\psi \in FL(\phi)$, ergo, if $\psi \in FLR(\phi)$. □

In the following lemma, $\#X$ denotes the cardinality of set X and $|\phi|$ denotes the length (number of symbols) of ϕ, excluding parentheses.

Lemma 3 *For any ϕ, $\#FLR(\phi) \leq (2 \times |\phi|)$*

Proof. By Lemma 6.3. of (Harel et al., 2000, ch. 6.1), $\#FL(\phi) \leq |\phi|$. The worst-case scenario is that no $\psi \in FL(\phi)$ begins with a '$[\star]$'. In that case, $\#FLR(\phi) = 2 \times \#FL(\phi)$. □

In what follows, a *non-standard* Routley model is a non-standard dynamic model in the sense of (Harel et al., 2000, ch. 6.3), where $R(\star)$ is serial, symmetric and functional. A *standard* Routley model is just a Routley model. M^ϕ for non-standard M is defined just as in the case of standard M. It is plain that M^ϕ is standard even if M is non-standard.

Definition 2 *Let M be a (standard or non-standard) Routley model. Define M^{ϕ}, the filtration of M trough ϕ, as follows:*

- $x \equiv_{\phi} y$ *iff for all* $\psi \in FLR(\phi)$, *(*$x \models \psi$ *iff* $y \models \psi$*)*
- $[x] = \{y \mid x \equiv_{\phi} y\}$
- $S^{\phi} = \{[x] \mid x \in S\}$
- $R^{\phi}_{a} = \{\langle [x], [y] \rangle \mid R_a xy\}$ *for all* $a \in AP$
- $V^{\phi}(p) = \{[x] \mid x \in V(p)\}$ *for all* $p \in AF$

R^{ϕ} for compound programs and $\models^{\phi}$ are defined on S^{ϕ} as in the definition of Routley models.

Lemma 4 *$R^{\phi}_{\star}$ is serial, symmetric and functional.*

Proof. Seriality and symmetry are straightforward. Functionality needs a bit more work. Assume that (a) $R^{\phi}_{\star}[x][y]$, $R^{\phi}_{\star}[x][z]$, but (b) $y \not\equiv_{\phi} z$. (b) means that there is $\psi \in FLR(\phi)$ such that $y \models \psi$ and $z \not\models \psi$. (a) means that there are x', x'', y', z' such that

- $x', x'' \in [x]$, $y' \in [y]$ and $z' \in [z]$
- $R_{\star}x'y'$ and $R_{\star}x''z'$

Together with (b), this means that

- $y' \models \psi$ and, therefore, $x' \models \langle \star \rangle \psi$, i.e. (Corollary 1), $x' \models [\star]\psi$
- $z' \not\models \psi$ and, therefore, $x'' \not\models [\star]\psi$

These two claims imply that $[\star]\psi \notin FLR(\phi)$ even if $\psi \in FLR(\phi)$. Now, either $\psi = [\star]\chi$ for some χ or not. If not, then $\psi \in FL(\phi)$ and, by the definition of FLR, $[\star]\psi \in FLR(\phi)$, a contradiction. If $\psi = [\star]\chi$ for some χ, then we can reason as follows. By symmetry of $R_{\star}$, $y' \models \psi$ implies $x' \models \chi$. By Lemma 2, $\chi \in FLR(\phi)$ and, hence, $x'' \models \chi$. However, $x'' \not\models [\star][\star]\chi$ ($x'' \not\models [\star]\psi$) implies, by Corollary 1, that $x'' \not\models \chi$. A contradiction. $\square$

Lemma 4 implies that M^{ϕ} is a standard Routley model, even if M happens to be non-standard.

Lemma 5 (Filtration Lemma) *Let M be a (standard or non-standard) Routley model. Then*

1. *For all* $\psi \in FLR(\phi)$, $x \models \psi$ *iff* $[x] \models^{\phi} \psi$

2. *For all* $[\alpha]\psi \in FLR(\phi)$,

   (a) *If* $R_{\alpha}xy$, *then* $R^{\phi}_{\alpha}[x][y]$

   (b) *If* $R^{\phi}_{\alpha}[x][y]$ *and* $x \models [\alpha]\psi$, *then* $y \models \psi$

Proof. The proof is an exact copy of the proofs of Lemmas 6.4 and 6.6. (Harel et al., 2000, ch. 6.2 and 6.3). As an inspection of the original proofs shows, this copy-pasting is possible because of Lemmas 1 and 2. □

Theorem 8 *If ϕ is satisfied in a (standard or non-standard) Routley model, then it is satisfied in some standard Routley model with at most $2^{(2\times|\phi|)}$ states.*

Proof. Lemma 5 implies that $(M, x) \models \phi$ for any (standard or non-standard) M only if $(M^{\phi}, [x]) \models^{\phi} \phi$. M^{ϕ} is a standard Routley model by Lemma 4. Finally, S^{ϕ} has no more states than the number of truth assignments to formulas in $FLR(\phi)$, which is by Lemma 3 at most $2^{(2\times|\phi|)}$. □

Theorem 6 follows immediately.

To prove Theorem 7, it is sufficient to establish the following claim; the theorem then follows by the standard argument (Harel et al., 2000, ch. 7). The $H(\mathsf{NPDL})$-canonical model M^c is defined in the usual way; $X \models^c \phi$ is defined as $\phi \in X$.

Lemma 6 *M^c is a non-standard Routley model.*

Proof. The fact that M^c is non-standard ($R^c(\alpha^*)$ is a superset of, not necessarily identical to, the reflexive transitive closure of $R(\alpha)$) is established as in PDL, as are the facts that $R^c(\alpha)$ for compound α and $\models^c$ behave as they should. The only new thing to prove is that $R^c(\star)$ is serial, symmetric and functional.

This follows from including the $\star$-axioms by standard modal reasoning. We formulate only the argument concerning functionality explicitly. Assume that $R^c_{\star}XY$ and $R^c_{\star}XZ$ but $Y \neq Z$. Hence, there is $\phi \in Y$ and $\phi \notin Z$. Consequently, $\neg\phi \in Z$, $\langle\star\rangle\phi \wedge \langle\star\rangle\neg\phi \in X$. It follows that $\langle\star\rangle\bot \in X$ ($\langle\star\rangle\chi \leftrightarrow [\star]\chi$ is an axiom). But $R^c_{\star}$ is serial and, therefore, $\bot \in Y$ for some maximal consistent Y. A contradiction. □

References

Arieli, O., & Avron, A. (1996). Reasoning with logical bilattices. *Journal of Logic, Language and Information*, *5*(1), 25–63.

Aucher, G., & Bolander, T. (2013). Undecidability in epistemic planning. In F. Rossi (Ed.), *Proceedings of the twenty-third international joint conference on artificial intelligence* (pp. 27–33). AAAI Press.

Baltag, A., & Moss, L. S. (2004). Logics for Epistemic Programs. *Synthese*, *139*(2), 165–224.

Baltag, A., & Smets, S. (2006). LQP: the dynamic logic of quantum information. *Mathematical Structures in Computer Science*, *16*(03), 491.

Belnap, N. (1977a). How a computer should think. In G. Ryle (Ed.), *Contemporary aspects of philosophy.* Oriel Press Ltd.

Belnap, N. (1977b). A useful four-valued logic. In J. M. Dunn & G. Epstein (Eds.), *Modern uses of multiple-valued logic* (pp. 5–37). Dordrecht: Springer Netherlands.

Bergfeld, J. M., & Sack, J. (2015). Deriving the correctness of quantum protocols in the probabilistic logic for quantum programs. *Soft Computing*.

Bolander, T., & Andersen, M. B. (2011). Epistemic planning for singleand multi-agent systems. *Journal of Applied Non-Classical Logics*, *21*(1), 9–34.

Dunn, J. M. (1976). Intuitive semantics for first-degree entailments and "coupled trees". *Philosophical Studies*, *29*, 149–168.

Dunn, J. M. (1993). Star and Perp: Two Treatments of Negation. *Philosophical Perspectives*, *7*, 331–357.

Fagin, R., Halpern, J. Y., & Vardi, M. (1995). A nonstandard approach to the logical omniscience problem. *Artificial Intelligence*, *79*(2), 203–240.

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

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated Planning: Theory and Practice*. San Francisco: Morgan Kaufmann.

Harel, D., Kozen, D., & Tiuryn, J. (2000). *Dynamic Logic*. MIT Press.

Odintsov, S., & Wansing, H. (2010). Modal logics with Belnapian truth values. *Journal of Applied Non-Classical Logics*, *20*(3), 279–301.

Rosenschein, S. J. (1981). Plan synthesis: A logical perspective. In *IJCAI'81 Proceedings of the 7th International Joint Conference on Artificial Intelligence* (pp. 331–337). San Franscisco: Morgan Kaufmann

Publishers Inc.
Routley, R., & Routley, V. (1972). The semantics of First Degree Entailment. *Noûs*, *6*(4), 335–359.
Sedlár, I. (2016). Propositional dynamic logic with Belnapian truth values. In L. Beklemishev, S. Demri, & A. Máté (Eds.), *Advances in Modal Logic. Vol. 11* (pp. 503–519). London: College Publications.
Stephan, W., & Biundo, S. (1993). A new logical framework for deductive planning. In *IJCAI'93 Proceedings of the 13th International Joint Conference on Artificial Intelligence* (pp. 32–38). San Francisco: Morgan Kaufmann Publishers Inc.
Teheux, B. (2014). Propositional dynamic logic for searching games with errors. *Journal of Applied Logic*, *12*(4), 377–394.

Igor Sedlár
Institute of Computer Science, Czech Academy of Sciences
The Czech Republic
E-mail: sedlar@cs.cas.cz