A reduction from DLP to PDL

Stéphane Demri

A reduction from DLP to PDL

Stéphane Demri
Laboratoire Spécification et Vérification
CNRS & INRIA Futurs projet SECSI & ENS Cachan
61, av. Pdt. Wilson, 94235 Cachan Cedex, France
email: demri@lsv.ens-cachan.fr

June 22, 2005

Abstract

We present a reduction from a new logic extending van der Meyden's dynamic logic of permission (DLP) into propositional dynamic logic (PDL), providing a 2EXPTIME decision procedure and showing that all the machinery for PDL can be reused for reasoning about dynamic policies. As a side-effect, we establish that DLP is EXPTIME-complete. The logic we introduce extends the logic DLP so that the policy set can be updated depending on its current value and such an update corresponds to add/delete transitions in the model, showing similarities with van Benthem's sabotage modal logic.

Key-words: dynamic logic of permissions, logic of programs, deletion of states/transitions, computational complexity.

1 Introduction

Reasoning about policies. Deontic logic is commonly defined as the logic of obligation, prohibition and permission. Indeed, reasoning about ideal and actual behaviors is useful in many fields of computer science, for instance to specify desired user behaviors, security policies, and normative integrity constraints, to quote a few examples [WM94]. From a formal viewpoint, numerous deontic logics are modal logics with extra features, some of them being quite original. For instance, in the possible-worlds semantics of such logics, it is common to distinguish the permitted states from the forbidden ones or alternatively the permitted actions from the forbidden ones. Moreover, several deontic logics [Mey88, vdM96, Bro03, PW04] are variants of propositional dynamic logic (PDL), see e.g., [HKT00], formalism sometimes used to specify the behavior of finite-state systems [CS91].

Among these logics, van der Meyden's dynamic logic of permission (DLP) [vdM96] can be defined as an extension of test-free PDL with modal operators that takes into account that some transitions are permitted (the green ones) and all the other ones are forbidden (the red ones). In a model, the set of green transitions forms the so-called policy set and modal operators in the logical language have semantics that distinguish green transitions from red ones. Reasoning about the permission of sequential actions has motivated the introduction of DLP [vdM96] in order to improve Meyer's logic [Mey88] which distinguishes permitted states from forbidden ones (instead permitted transitions from forbidden ones as in DLP). Hilbert-style axiomatization of DLP is provided in [vdM96] and an NEXPTIME upper bound for the satisfiability problem is established (even though no complexity questions are explicitly discussed in that paper). In [PW04], DLP is generously extended in order to specify in the logical language the updates of the policy set by adding or by deleting transitions. This is a very substantial extension of DLP and in [PW04] an axiomatization is provided as well as an NEXPTIME upper bound (the proofs of these results are promised for the full version of [PW04]). The ability to add or delete transitions is reminiscent to van Benthem's sabotage modal logic SML [vB02] whose satisfiability problem for a variant has been proved undecidable in [LR03] (when deleting transitions instead of deleting states). The destructive dimension of SML and its ability to quantify over alternative models (obtained by deleting one transition/state) seem to be the main reasons for its undecidability. Hence, the decidability of the logic $\mathrm{DLP}_{\mathrm{dyn}}$ introduced in [PW04] is a quite remarkable result. In order to grasp the different decidability status of SML and $\mathrm{DLP}_{\mathrm{dyn}}$, in SML there is no way to specify anything about which states or transitions are deleted. By contrast, in $\mathrm{DLP}_{\mathrm{dyn}}$, the object language can specify the transitions to be deleted or added.

Motivations. However, by experience, we know that many logics have been able to be translated into PDL even though they were introduced for their own sake: epistemic logics [FI87], deontic logics [Bro03], description logics [Sch91, dGL94], information logics [DG00], regular grammar logics [Dem01] and agent dynamic logics [STH04]. The list can be easily augmented and the main motivation of this work is to try to translate $\mathrm{DLP}_{\mathrm{dyn}}$ into standard PDL. Even if at first glance, this bet does not sound very reasonable because of the features of $\mathrm{DLP}_{\mathrm{dyn}}$ [PW04], a lot can be gained. Indeed, the existence of such an hypothetical semantical translation would explain why standard proof techniques worked smoothly for dynamic logics of permission [vdM96, PW04]. More importantly, several proof methods and theoretical results for PDL, see e.g. [VW86], would apply immediately to these logics.

Our contribution. We embed an extension of $\mathrm{DLP}_{\mathrm{dyn}}$, namely $\mathrm{DLP}^{+}_{\mathrm{dyn}}$, into standard PDL by encoding faithfully $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ semantics and by taking advantage of various fundamental properties of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$. Roughly speaking, we allow in $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ the test operator "?" and the operators for updating the policy set can be parameterized by the current policy set, a novelty with respect to [PW04]. In spite of these substantial extensions, the exponential-time translation from $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ into PDL entails that $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ satisfiability is decidable in 2EXPTIME. As a corollary, we also get

- $*$ the EXPTIME-completeness of DLP satisfiability,
- $*$ the EXPTIME-completeness of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ restricted to formulae of change depth (to be defined) at most k, for some fixed $k \geq 0$,

which are all new results. It is worth observing that the extension we have introduced is not primarily motivated by the need to increase the expressive power of $\mathrm{DLP}_{\mathrm{dyn}}$, but rather for technical reasons. In general, in the paper we focus our attention on the reductions rather than on interpretations of the concepts from the deontic viewpoint. Such interpretations can be found in the original papers, see e.g. [Mey88, vdM96, PW04].

In [vdM96], a strong motivation to introduce the logic DLP is to replace the concept of permitted states from [Mey88] by the concept of permitted actions. In Section 4, we define a variant of Meyer's logic [Mey88] with the ability to update dynamically the interpretation of the violation constant vc as $\mathrm{DLP}_{\mathrm{dyn}}$ [PW04] is the dynamic counterpart of DLP [vdM96]. The violation constant vc is defined as a distinguished propositional variable for which no action can lead to a state satisfying it. We show that this new logic also admits a translation into PDL.

All our proofs are semantical in nature and do not rely on sophisticated completeness proofs as those for PDL-like logics, see e.g. [vdM96, PW04], which allows us to have quite elementary proofs.

Plan of the paper. In Section 2, we introduce the logic $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ extending the logic $\mathrm{DLP}_{\mathrm{dyn}}$ [PW04], the fragments considered in the paper and we present the main difficulties to deal with $\mathrm{DLP}^{+}_{\mathrm{dyn}}$. In Section 3, we define the translation from $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ into PDL and we show its soundness. Complexity issues are also discussed in this section. In Section 4, we introduce a counterpart of Meyer's logic with update of the set of forbidden bad states. Section 5 presents concluding remarks and open problems.

2 The logics PDL, DLP, $\mathbf{DLP}_{\mathrm{dyn}}$ and $\mathbf{DLP}^{+}_{\mathrm{dyn}}$

2.1 The logic for dynamic policies $\mathrm{DLP}^{+}_{\mathrm{dyn}}$

Given a set $\Pi_0 = \{a_i : i \geq 1\}$ of atomic actions and a set $\mathrm{PROP} = \{p_i : i \geq 1\}$ of propositional variables, we define the set Π of action expressions and the set FOR of formulae for $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ inductively as follows:

$$\Pi \ni \alpha, \beta ::= a_i \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid \phi?$$

$$\begin{aligned}\mathrm{FOR} \ni \phi, \psi ::= {} & p_i \mid \phi \wedge \psi \mid \neg\phi \mid [\alpha]\phi \mid \mathtt{perm}(\alpha)\phi \mid \\ & \mathtt{freeperm}(\alpha)\phi \mid \mathtt{grant}(\psi, \psi')\phi \mid \mathtt{revoke}(\psi, \psi')\phi.\end{aligned}$$

As in propositional dynamic logic PDL (see e.g. [HKT00]) we have an countably infinite supply of atomic actions and propositional variables, but a given formula/action expression contains only a finite amount of such syntactic objects. Given an action expression α, we write $\mathrm{L}(\alpha)$ to denote the regular language of α over the finite alphabet composed of the atomic actions together with action expressions of the form $\phi?$ occurring in α. A $\mathrm{DLP}^{+}_{\mathrm{dyn}}$-model $\mathcal{M}$ is a structure of the form $\langle W, (R_a)_{a\in\Pi_0}, V, P\rangle$ where

- W is a non-empty set of states. Each state represents the current configuration of an application in time.
- $(R_a)_{a\in\Pi_0}$ is a family of binary relations over W. Elements of R_a are transitions between the states that correspond to the progress of the application.
- $V : W \to \mathcal{P}(\mathrm{PROP})$ is the meaning function that specifies which atomic propositions hold true in each state.
- $P \subseteq W \times W$ is a binary relation representing a policy set, i.e. the set of permitted transitions.

We say that the formula ϕ is satisfied in the model $\mathcal{M}$ by the state s (written $\mathcal{M}, s \models \phi$) if the following conditions are satisfied:

- $\mathcal{M}, s \models p \overset{\text{def}}{\Leftrightarrow} p \in V(s)$ for every $p \in \mathrm{PROP}$,
- $\mathcal{M}, s \models \phi \wedge \psi \overset{\text{def}}{\Leftrightarrow} \mathcal{M}, s \models \phi$ and $\mathcal{M}, s \models \psi$,
- $\mathcal{M}, s \models \neg\phi \overset{\text{def}}{\Leftrightarrow}$ not $\mathcal{M}, s \models \phi$,
- $\mathcal{M}, s \models [\alpha]\phi \overset{\text{def}}{\Leftrightarrow}$ for all paths of the form $s_0 \overset{A_0}{\to} s_1 \overset{A_1}{\to} \cdots \overset{A_{n-1}}{\longrightarrow} s_n$ in $\mathcal{M}$ with $A_0 A_1 \cdots A_{n-1} \in \mathrm{L}(\alpha)$ and $s_0 = s$, we have $\mathcal{M}, s_n \models \phi$,

* $\mathcal{M}, s \models \texttt{perm}(\alpha)\phi \stackrel{\text{def}}{\Leftrightarrow}$ there is a path of the form $s_0 \stackrel{A_0}{\rightarrow} s_1 \stackrel{A_1}{\rightarrow} \cdots \stackrel{A_{n-1}}{\longrightarrow} s_n$ in $\mathcal{M}$ with $A_0 A_1 \cdots A_{n-1} \in \mathrm{L}(\alpha)$ such that $s_0 = s$, for every $i \in \{0, \ldots, n-1\}$, $A_i \in \Pi_0$ implies $\langle s_i, s_{i+1} \rangle \in P$, and $\mathcal{M}, s_n \models \phi$. Such a path is said to be P-green. A P-green corresponds to a legal sequence of transitions.

* $\mathcal{M}, s \models \texttt{freeperm}(\alpha)\phi \stackrel{\text{def}}{\Leftrightarrow}$ there is no path $s_0 \stackrel{A_0}{\rightarrow} s_1 \stackrel{A_1}{\rightarrow} \cdots \stackrel{A_{n-1}}{\longrightarrow} s_n$ in $\mathcal{M}$ such that

  - $A_0 A_1 \cdots A_{n-1} \in \mathrm{L}(\alpha)$ and $s_0 = s$,
  - for some $i \in \{0, \ldots, n-1\}$, $A_i \in \Pi_0$ and $\langle s_i, s_{i+1} \rangle \notin P$,
  - $\mathcal{M}, s_n \models \phi$.

  Such a path is said to be P-red. A path is then P-red if one of its transitions is non-legal.

* $\mathcal{M}, s \models \texttt{grant}(\psi, \psi')\ \phi \stackrel{\text{def}}{\Leftrightarrow} \langle W, (R_a)_{a \in \Pi_0}, V, P \cup P^{\psi,\psi'}_{\mathcal{M}} \rangle, s \models \phi$ where $P^{\psi,\psi'}_{\mathcal{M}} = \{\langle t, t' \rangle \in W^2 : \mathcal{M}, t \models \psi \text{ and } \mathcal{M}, t' \models \psi'\}$,

* $\mathcal{M}, s \models \texttt{revoke}(\psi, \psi')\ \phi \stackrel{\text{def}}{\Leftrightarrow} \langle W, (R_a)_{a \in \Pi_0}, V, P \setminus P^{\psi,\psi'}_{\mathcal{M}} \rangle, s \models \phi$.

An intuition explanation of the operator $\texttt{grant}(\psi, \psi')\phi$ is as follows: ϕ holds (as a norm) under the condition that all ψ, ψ'-transitions are granted. With this reading, granting and revoking is not really about 'updating'. The operators are more like conditional operators.

We recall that in the above definition, $t \stackrel{\phi?}{\rightarrow} t'$ iff $t = t'$ and $\mathcal{M}, t \models \phi$. We use the abbreviation $\langle \alpha \rangle \phi$ for $\neg[\alpha]\neg\phi$. In a model, we write R_α to denote the binary relation

$$\{\langle s, t \rangle : s_0 \stackrel{A_0}{\rightarrow} s_1 \stackrel{A_1}{\rightarrow} \cdots \stackrel{A_{n-1}}{\longrightarrow} s_n,\ A_0 A_1 \cdots A_{n-1} \in \mathrm{L}(\alpha),\ s = s_0, t = s_n\}.$$

As usual, $\phi \in \mathrm{DLP}^{+}_{\mathrm{dyn}}$ is satisfiable iff there is a $\mathrm{DLP}^{+}_{\mathrm{dyn}}$-model $\mathcal{M}$ and a state s in $\mathcal{M}$ such that $\mathcal{M}, s \models \phi$.

$\texttt{perm}(\alpha)\phi$ corresponds to $\Diamond(\alpha, \phi)$ in [vdM96] and $\texttt{freeperm}(\alpha)\phi$ corresponds to $\pi(\alpha, \phi)$. Hence, we have adopted the notation from [PW04] since we also use the operators `grant` and `revoke` from [PW04] that are not present in [vdM96]. Motivations and explanations about P-green and P-red paths can be found in [vdM96, PW04] whereas numerous examples of deontic properties expressible in $\mathrm{DLP}_{\mathrm{dyn}}$ (and therefore in $\mathrm{DLP}^{+}_{\mathrm{dyn}}$) can be found in [PW04].

In Fig. 2.1, we illustrate the semantics on a simple model. In the double circled state, $\texttt{freeperm}(a; d)r$ does not hold because the unique path labelled by $a \cdot d$ starting at the double circled state is P-red. By contrast, $\texttt{grant}(q, r)\texttt{freeperm}(a; d)r$

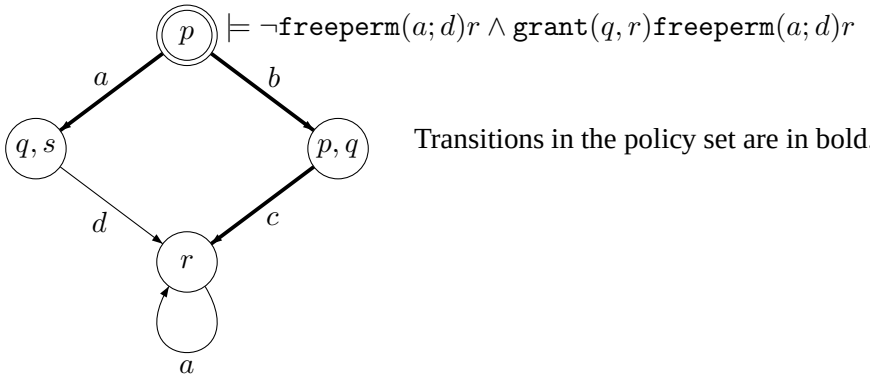

Figure 1: Illustration of the semantics

holds true on that state, because the effect of $\texttt{grant}(q, r)$ is to include $\{q, s\} \xrightarrow{d} \{r\}$ in the new value of the policy set.

2.2 Known fragments of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$

The logic $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ has been designed to contain all the logics we need in the paper. The logic PDL is equivalent to the fragment of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ restricted to formulae without any of the four operators dealing with policies. The main result of the paper is to define an exponential-time reduction from $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ into PDL, providing not only a 2EXPTIME upper bound for the satisfiability problem, but also showing that our extended logic $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ is not more expressive than PDL. Satisfiability for PDL is known to be EXPTIME-complete [FL79, Pra79].

The logic $\mathrm{DLP}_{\mathrm{dyn}}$ [PW04] is the fragment of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ restricted to formulae without the test operator "?" and with the `grant` and `revoke` operators restricted to propositional formulae in their first two arguments. [PW04] states that satisfiability for $\mathrm{DLP}_{\mathrm{dyn}}$ is in NEXPTIME. Unlike $\mathrm{DLP}_{\mathrm{dyn}}$, the change of policy in $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ may depend also on the current policy set since there are no restrictions on the first two arguments of `grant` and `revoke`. Numerous illustrations of the use of $\mathrm{DLP}_{\mathrm{dyn}}$ from the deontic viewpoint can be found in [PW04].

Redefining DLP [vdM96] from $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ requires a bit more care. The language of the logic DLP is the language of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ restricted to formulae without the test operator, `grant` and `revoke`. However, as it was defined initially in [vdM96], the DLP models are also a bit different: the relations R_a are defined in terms of finite sequences instead of sequences of length 1 in $\mathrm{DLP}^{+}_{\mathrm{dyn}}$. More precisely, a DLP model $\mathcal{M}$ is a structure of the form $\langle W, (X_a)_{a \in \Pi_0}, V, P \rangle$ where W is a non-empty set, $V : W \to \mathcal{P}(\mathrm{PROP})$, $P \subseteq W \times W$ is a binary relation representing a policy

set, and $(X_a)_{a\in\Pi_0}$ is a family of sequences of the form $\overbrace{s_0 \to s_1 \to \cdots \to s_n}^{a} \in X_a$ with $n \geq 0$ and $s_0, \ldots, s_n \in W$. This last point is the only difference with the notion of $\mathrm{DLP}^+_{\mathrm{dyn}}$ models. The definition of the satisfaction relation is also modified accordingly:

* $\mathcal{M}, s \models [\alpha]\phi \overset{\text{def}}{\Leftrightarrow}$ for all paths of the form

$$\overbrace{s_0^0 \to \cdots \to s_0^{n_0}}^{A_0}, \overbrace{s_1^0 \to \cdots \to s_1^{n_1}}^{A_1}, \ldots, \overbrace{s_m^0 \to \cdots \to s_m^{n_m}}^{A_m}$$

with $s_i^{n_i} = s_{i+1}^0$ for every $0 \leq i < m$, $A_0 A_1 \cdots A_m \in \mathrm{L}(\alpha)$ and $s_0^0 = s$, we have $\mathcal{M}, s_m^{n_m} \models \phi$.

* $\mathcal{M}, s \models \mathtt{perm}(\alpha)\phi \overset{\text{def}}{\Leftrightarrow}$ there is a path

$$\overbrace{s_0^0 \to \cdots \to s_0^{n_0}}^{A_0}, \overbrace{s_1^0 \to \cdots \to s_1^{n_1}}^{A_1}, \ldots, \overbrace{s_m^0 \to \cdots \to s_m^{n_m}}^{A_m}$$

with $s_i^{n_i} = s_{i+1}^0$ for every $0 \leq i < m$, $A_0 A_1 \cdots A_m \in \mathrm{L}(\alpha)$ such that $s_0^0 = s$, for all $i \in \{0, \ldots, m\}$ and $j \in \{0, \ldots, n_i - 1\}$, $A_i \in \Pi_0$ implies $\langle s_i^j, s_i^{j+1}\rangle \in P$, and $\mathcal{M}, s_m^{n_m} \models \phi$. By imposing $n_0 = \cdots = n_m = 1$, we regain the $\mathrm{DLP}^+_{\mathrm{dyn}}$ semantics.

* $\mathcal{M}, s \models \mathtt{freeperm}(\alpha)\phi \overset{\text{def}}{\Leftrightarrow}$ there is no path

$$\overbrace{s_0^0 \to \cdots \to s_0^{n_0}}^{A_0}, \overbrace{s_1^0 \to \cdots \to s_1^{n_1}}^{A_1}, \ldots, \overbrace{s_m^0 \to \cdots \to s_m^{n_m}}^{A_m}$$

such that

  - $s_i^{n_i} = s_{i+1}^0$ for every $0 \leq i < m$,
  - $A_0 A_1 \cdots A_m \in \mathrm{L}(\alpha)$,
  - $s_0^0 = s$,
  - for some $i \in \{0, \ldots, m\}$ and $j \in \{0, \ldots, n_i - 1\}$, $A_i \in \Pi_0$ and $\langle s_i^j, s_i^{j+1}\rangle \notin P$,
  - $\mathcal{M}, s_m^{n_m} \models \phi$.

Fortunately, the slight difference in the semantics of DLP and $\mathrm{DLP}^+_{\mathrm{dyn}}$ does not affect the satisfiability of DLP formulae.

Lemma 2.1. For any DLP formula ϕ, ϕ is satisfiable with the DLP semantics iff ϕ is satisfiable with the $\mathrm{DLP}^+_{\mathrm{dyn}}$ semantics.

The proof below is purely semantical.

Proof: Obviously ϕ is $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ satisfiable implies ϕ is DLP satisfiable (sequences of length 1 are particular sequences of arbitrary finite length).

Now suppose that ϕ is DLP satisfiable and $b_1, \ldots, b_N$ are the atomic actions occurring in ϕ. There is a DLP model $\mathcal{M} = \langle W, (X_a)_{a \in \Pi_0}, V, P \rangle$ and $s_0 \in W$ such that $\mathcal{M}, s_0 \models \phi$. Let us build a $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ model $\mathcal{M}' = \langle W', (R_a)_{a \in \Pi_0}, V', P' \rangle$ by unfolding the model $\mathcal{M}$ in the following way:

- W' defined is the set of finite non-empty sequences of the form

$$\overbrace{s_0^0 \to \cdots \to s_0^{n_0}}^{b_{i_0}}, \cdots, \overbrace{s_m^0 \to \cdots \to s_m^{n_m}}^{b_{i_m}}$$

  with $s_i^{n_i} = s_{i+1}^0$ for every $0 \leq i < m$, $b_{i_0}, \ldots, b_{i_m} \subseteq \{b_1, \ldots, b_N\}$ and $s_0^0 = s_0$. We also add $\overbrace{s_0 \to s_0}^{b_{new}}$ to W' where b_{new} is an atomic action not in $\{b_1, \ldots, b_N\}$.

- $$\overbrace{s_0^0 \to \cdots \to s_0^{n_0}}^{b_{i_0}}, \ldots, \overbrace{s_m^0 \to \cdots \to s_m^{n_m}}^{b_{i_m}} \in R'_{b_i}$$

$$\overbrace{s_0^0 \to \cdots \to s_0^{n_0}}^{b_{i_0}}, \ldots, \overbrace{s_m^0 \to \cdots \to s_m^{n_m}}^{b_{i_m}}, \overbrace{s_{m+1}^0 \to \cdots \to s_{m+1}^{n_{m+1}}}^{b_i}$$

  for every $i \in \{1, \ldots, N\}$ with $s_i^{n_i} = s_{i+1}^0$ for $0 \leq i \leq m$,

- $V'(\overbrace{s_0^0 \to \cdots \to s_0^{n_0}}^{b_{i_0}}, \ldots, \overbrace{s_m^0 \to \cdots \to s_m^{n_m}}^{b_{i_m}}) \stackrel{\text{def}}{=} V(s_m^{n_m})$,

- $$\overbrace{s_0^0 \to \cdots \to s_0^{n_0}}^{b_{i_0}}, \ldots, \overbrace{s_m^0 \to \cdots \to s_m^{n_m}}^{b_{i_m}} \in P'$$

$$\overbrace{s_0^0 \to \cdots \to s_0^{n_0}}^{b_{i_0}}, \ldots, \overbrace{s_m^0 \to \cdots \to s_m^{n_m}}^{b_{i_m}}, \overbrace{s_{m+1}^0 \to \cdots \to s_{m+1}^{n_{m+1}}}^{b_i}$$

  $\stackrel{\text{def}}{\Leftrightarrow}$ for every $0 \leq k \leq n_{m+1} - 1$, $\langle s_{m+1}^k, s_{m+1}^{k+1} \rangle \in P$.

One can show that $\mathcal{M}, s_0 \models \phi$ iff $\mathcal{M}', \overbrace{s_0 \rightarrow s_0}^{b_{new}} \models \phi$ by induction on the structure of formulae. $\Box$

Hence, DLP can be really viewed as a proper fragment of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$. We write $cd(\phi)$ to denote the change depth of the formula ϕ defined as the maximal imbrication of operators dealing with updates of policies in ϕ. For instance,

$$cd(\texttt{revoke}(p, \texttt{grant}(q_1, q_2)\ p)\ q_1) = 2.$$

DLP is simply the restriction of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ to formulae of change depth 0 and with no occurrence of the test operator. For the sake of clarity let us define formally the change depth of formulae and action expressions:

- $cd(p) = 0$, $cd(\neg\phi) = cd(\phi)$, $cd(\phi_1 \wedge \phi_2) = max(cd(\phi_1), cd(\phi_2))$,
- $cd([\alpha]\phi) = cd(\texttt{perm}(\alpha)\phi) = cd(\texttt{freeperm}(\alpha)\phi)) = cd(\alpha) + cd(\phi)$,
- $cd(\texttt{revoke}(\psi_1, \psi_2)\phi) = 1 + max(cd(\psi_1), cd(\psi_2)) + cd(\phi)$,
- $cd(\texttt{grant}(\psi_1, \psi_2)\phi) = cd(\texttt{revoke}(\psi_1, \psi_2)\phi)$,
- $cd(\alpha; \beta) = cd(\alpha \cup \beta) = max(cd(\alpha), cd(\beta))$,
- $cd(\phi?) = cd(\phi)$, $cd(\alpha^*) = cd(\alpha)$.

2.3 What makes $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ difficult to handle

The logic $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ has various features that make its decidability status difficult to establish and its translation into PDL improbable at first glance.

2.3.1 Presence of intersection

The semantics of the operator `perm` can be rephrased in PDL by replacing $\texttt{perm}(\alpha)$ by $\langle t^{\forall \mathrm{g}}(\alpha)\rangle$ where $t^{\forall \mathrm{g}}(\alpha)$ is obtained from α by replacing each atomic action a by $a \cap \alpha_P$. The operator $\cap$ is interpreted as the relation intersection and α_P is an action expression whose interpretation is the policy set P. However, PDL with intersection [Dan84] has been proved decidable in 2EXPTIME, see also [Lut05]. At first glance, it is worth observing that the PDL-like deontic logic introduced in [Mey88] has also the intersection operator $\cap$, but not the Kleene star operator *.

2.3.2 Presence of complement

However, pursuing the encoding into PDL in this way, we need also complementation. Indeed, the semantics of the operator `freeperm` can be rephrased in PDL by

replacing $\texttt{freeperm}(\alpha)$ by $\neg\langle t^{\exists r}(\alpha)\rangle$. The intended meaning of $\langle t^{\exists r}(\alpha)\rangle\top$ is that there is a path labelled by a word in $\mathrm{L}(\alpha)$ such that at least one transition labelled by an atomic action does not belong to P, i.e. this transition does not belong to the interpretation of α_P. The map $t^{\exists r}$ is defined as follows:

* $t^{\exists r}(\phi?) = \bot?$. Indeed, the witness red transition has to be labelled by a letter in Π_0 which is not the case when the transition is labelled by $\phi?$.
* $t^{\exists r}(\alpha_1 \cup \alpha_2) = t^{\exists r}(\alpha_1) \cup t^{\exists r}(\alpha_2)$,
* $t^{\exists r}(\alpha_1;\alpha_2) = t^{\exists r}(\alpha_1);\alpha_2 \cup \alpha_1;t^{\exists r}(\alpha_2)$. The red transition is either in the α_1-part of $\alpha_1;\alpha_2$ or in the α_2-part if not in the α_1-part.
* $t^{\exists r}(\alpha^*) = \alpha^*;t^{\exists r}(\alpha);\alpha^*$. The red transition is in one specific α-part.
* $t^{\exists r}(a) = a \cap -\alpha_P$.

$t^{\exists r}$ is intended to enforce the presence of a (red) transition not in P. However, PDL with complement is undecidable (see e.g. [HKT00]) and PDL with negation of atomic programs shown in EXPTIME in [LW04] cannot capture both intersection and negation on atomic programs. An alternative way would be to encode `perm` and `freeperm` by decomposing each atomic action b as the union action expression $b^g \cup b^r$ (green part union with the red part) such that the following conditions hold in the models

(PROP1) R_{b^g} and R_{b^r} are disjoint,

(PROP2) for every $\langle s,t\rangle \in W \times W$, $\langle s,t\rangle \in R_{b^g}$ (hence $\langle s,t\rangle$ is intended to be in P) implies $\langle s,t\rangle \notin R_{b'^r}$ for every $b' \in \Pi_0$.

Unfortunately, none of these two conditions are modally definable. However, as we will see, disjointness can be imposed in the PDL models without changing the class of satisfiable formulae. Condition (PROP2) is a consequence of the fact that in the DLP semantics, the policy set P is defined as a set of pairs of states and not as a set of triples composed of two states and an atomic action.

2.3.3 Change of models

So far, the above points concern the DLP part of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$. The operators `grant` and `revoke` introduced in [PW04] force the interpretation of subformulae in an alternative model which is reminiscent to the destructive aspect of van Benthem's sabotage modal logic [vB02]. Indeed, sabotage modal logic (SML) defined in [vB02] admits formulae of the form

$$\phi ::= p \mid \neg\phi \mid \phi_1 \wedge \phi_1 \mid \Diamond\phi \mid \langle -\rangle\phi$$

and the models are Kripke structures of the form $\mathcal{M} = \langle W, R, V\rangle$ where $V : W \to \mathcal{P}(\mathrm{PROP})$. The only change with respect to standard possible-worlds semantics is the following: $\mathcal{M}, w \models \langle - \rangle \phi$ iff there is a $w' \in W$ such that $\mathcal{M}', w \models \phi$ where $\mathcal{M}'$ is the restriction of $\mathcal{M}$ to $W \setminus \{w'\}$. The decidability status for the satisfiability problem of SML is open. However, a variant of SML has been introduced in [LR03], we call it SML$'$ herein, for which the satisfiability problem has been proved undecidable [LR03]. Instead of deleting states in the models as in SML, SML$'$ provides the possibility to withdraw transitions, a feature also shared with $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ (but in a different fashion). Formulae of SML$'$ are of the form

$$\phi ::= p \mid \neg\phi \mid \phi_1 \wedge \phi_1 \mid \Diamond_a \phi \mid \langle - \rangle_a \phi$$

where a takes its value in a finite alphabet Σ. The models are Kripke structures of the form $\mathcal{M} = \langle W, (R_a)_{a\in\Sigma}, V\rangle$ and the only change with respect to standard possible-worlds semantics is the following: $\mathcal{M}, w \models \langle - \rangle_a \phi$ iff there is $\langle w', w''\rangle \in R_a$ such that $\mathcal{M}', w \models \phi$ where $\mathcal{M}'$ is obtained from $\mathcal{M}$ by simply withdrawing $\langle w', w''\rangle$ from R_a. The satisfiability problem for SML$'$ is shown undecidable in [LR03] as soon as $|\Sigma| \geq 2$ (another variant is shown undecidable in [Roh04] in which deletion of the transitions is done locally to the current state). In the case $|\Sigma| = 1$, the decidability status is open. Hence, both SML$'$ and $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ have primitives in the language to withdraw edges. Even worse, in $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ transitions can also be added to the policy set.

Hence, in view of the above points, it is not surprising that $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ is not an easy logic to study. However, in the following, we shall show that $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ can be translated into PDL by taking advantage of a few fundamental properties.

3 A purely semantical reduction

3.1 Fundamental properties of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$

In order to define the reduction from $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ into PDL, some preliminary remarks are needed that will help hopefully the comprehension of the translation.

3.1.1 How the policy set can be restricted

The definition of the satisfaction relation can be modified in such a way that for the semantics of $\mathtt{grant}(\psi_1, \psi_2)\ \phi$ and $\mathtt{revoke}(\psi_1, \psi_2)\ \phi$, we can restrict ourselves to the pairs in P' that belong to atomic relations occurring in ϕ (see Lemma 3.1 below).

Lemma 3.1. Let $\mathcal{M} = \langle W, (R_a)_{a\in\Pi_0}, V, P\rangle$ and $\mathcal{M}' = \langle W, (R_a)_{a\in\Pi_0}, V, P'\rangle$ be models such that $P' = P \cap (\bigcup_{1\leq i\leq N} R_{b_i})$ for some finite set of atomic actions $\{b_1, \ldots, b_N\}$. Then for every formula ϕ built over the atomic actions in $\{b_1, \ldots, b_N\}$, for every $s \in W$, $\mathcal{M}, s \models \psi$ iff $\mathcal{M}', s \models \psi$.

The proof is by an easy verification (structural induction on ψ). In the following, when $\mathcal{M}, s \models \phi$ without any loss of generality we can assume that $P \subseteq R_{b_1} \cup \cdots \cup R_{b_N}$ where $b_1, \ldots, b_N$ are atomic actions occurring in ϕ.

In Lemma 3.2 below, we show that if α_P is an action expression in PDL interpreted by $P \cap (R_{b_1} \cup \cdots \cup R_{b_N})$, then one can easily built an action expression in PDL interpreted by $(P \cup P_{\mathcal{M}}^{\psi,\psi'}) \cap (R_{b_1} \cup \cdots \cup R_{b_N})$ when ψ, ψ' are built over the atomic actions $b_1, \ldots, b_N$. In that way, we can deal with green transitions and red transitions with no explicit use of intersection and complement on relations.

Lemma 3.2. Let $\mathcal{M} = \langle W, (R_a)_{a\in\Pi_0}, V, P\rangle$ be a $\mathrm{DLP}^{+}_{\mathrm{dyn}}$-model and $\{b_1, \ldots, b_N\}$ be a finite set of atomic actions such that there are action expressions α_P and α_{-P} verifying

(I) $R_{\alpha_P} = P \cap (R_{b_1} \cup \cdots \cup R_{b_N})$,

(II) $R_{\alpha_{-P}} = ((W \times W) \setminus P) \cap (R_{b_1} \cup \cdots \cup R_{b_N})$.

Then, for all formulae ψ, ψ' built over $\{b_1, \ldots, b_N\}$,

(III) $(P \cup P_{\mathcal{M}}^{\psi,\psi'}) \cap (R_{b_1} \cup \cdots \cup R_{b_N}) = R_\beta$ with

$$\beta = \alpha_P \cup (\psi?; b_1; \psi'?) \cup \cdots \cup (\psi?; b_N; \psi'?)$$

(IV) $((W \times W) \setminus (P \cup P_{\mathcal{M}}^{\psi,\psi'})) \cap (R_{b_1} \cup \cdots \cup R_{b_N}) = R_{\beta'}$ with

$$\beta' = (\neg\psi)?; \alpha_{-P} \cup \alpha_{-P}; (\neg\psi')?$$

The policy sets from Lemma 3.2(III) and Lemma 3.2(IV) are typically obtained with the operators `grant` and `revoke`.

Proof: (III) can be easily shown by taking advantage of the equivalence between the propositions below:

* $s \xrightarrow{\psi?} s \xrightarrow{b_i} s' \xrightarrow{\psi'?} s'$ for some i,
* $\mathcal{M}, s \models \psi$, $\langle s, s'\rangle \in R_{b_i}$ for some i, and $\mathcal{M}, s' \models \psi'$,
* $\langle s, s'\rangle \in P_{\mathcal{M}}^{\psi,\psi'} \cap (R_{b_1} \cup \cdots \cup R_{b_N})$.

Similarly, (IV) can be easily shown by taking advantage of the equivalence between the propositions below:

* either $\langle s, s' \rangle \in R_{(\neg\psi)?;\alpha_{-P}}$ or $\langle s, s' \rangle \in R_{\alpha_{-P};(\neg\psi')?}$,
* (not ($\mathcal{M}, s \models \psi$ and $\mathcal{M}, s' \models \psi'$)) and $\langle s, s' \rangle \in R_{\alpha_{-P}}$,
* $\langle s, s' \rangle \in (W \times W) \setminus P^{\psi,\psi'}_{\mathcal{M}}$ and $\langle s, s' \rangle \in ((W \times W) \setminus P) \cap (R_{b_1} \cup \cdots \cup R_{b_N})$,
* $\langle s, s' \rangle \in ((W \times W) \setminus (P \cup P^{\psi,\psi'}_{\mathcal{M}})) \cap (R_{b_1} \cup \cdots \cup R_{b_N})$.

□

Hence, assuming that α_P is an action expression interpreted by $P \cap (\bigcup_{a\in\phi} R_a)$ and α_{-P} is an action expression interpreted by $(W \times W \setminus P) \cap (\bigcup_{a\in\phi} R_a)$, the program expression $\alpha_P \cup (\psi?; \cup_{a\in\phi}; \psi'?)$ is interpreted by $P \cup P'$ when dealing with $\texttt{grant}(\psi, \psi')\phi$. Similarly, in case of $\texttt{revoke}(\psi, \psi')\ \phi$, the program expression $(\neg\psi?; \alpha_{-P}) \cup (\alpha_{-P}; \neg\psi'?)$ is interpreted by $P \setminus P'$. This means that the new policy set (either by revoking or by granting) can be expressed in PDL assuming that its initial value could be interpreted by a program expression in PDL. In the forthcoming translation function, one argument is devoted to the P part and a second one is devoted to the complement of P both restricted to atomic actions present in the formula to translate. Observe that in PDL, complementation of relations is not present.

3.1.2 Disjointness of atomic actions

In order to encode the initial value of P by an action expression in PDL, we shall encode every atomic action b_i occurring in the formula ϕ to be translated by the union $b_i^g \cup b_i^r$ (initial green part and initial red part, respectively). Additionally, if $b_1, \ldots, b_N$ are the atomic actions present in ϕ, we shall enforce that $b_1^g, b_1^r, \ldots, b_N^g, b_N^r$ are interpreted by disjoint relations which is possible in PDL, see Lemma 3.3. Both properties (PROP1) and (PROP2) are satisfied in that case. In that way, the initial value of α_P can be represented by the action expression $b_1^g \cup \cdots \cup b_N^g$ and its complement $-\alpha_P$ (relatively to $\bigcup_{b\in\phi}(b^g \cup b^r)$) by the action expression $b_1^r \cup \cdots \cup b_N^r$. By applying a finite amount of `grant` and `revoke`, Lemma 3.2 guarantees that the current values of α_P and $-\alpha_P$ are equivalent to action expressions of the form $\psi_1?; \cdots; \psi_n?; b_i^\bullet; \psi'_1? \cdots; \psi'_{n'}?$ with $\bullet \in \{g, r\}$ (of course, not the same sets). $\psi_1, \cdots, \psi_n, \psi'_1, \psi'_{n'}$ are PDL formulae obtained by translation and n is not necessarily equal to n'.

3.1.3 Encoding transition relations

It remains to explain how b_i is treated in the translation process.

- When b_i occurs in the immediate context of $[\alpha]$, b_i is encoded by $b_i^g \cup b_i^r$.
- When b_i occurs in the immediate context of $\texttt{perm}(\alpha)$, b_i is encoded by the union of the program expressions of the form

  $$\psi_1?; \cdots ; \psi_n?; b_i^\bullet; \psi'_1? \cdots ; \psi'_{n'}?$$

  with $\bullet \in \{g, r\}$ occurring in α_P.
- Similarly, when b_i occurs in the context of $\texttt{freeperm}(\alpha)$, some occurrences of b_i are encoded by the union of the program expressions of the form $\psi_1?; \cdots ; \psi_n?; b_i^\bullet; \psi'_1? \cdots ; \psi'_{n'}?$ with $\bullet \in \{g, r\}$ occurring in $-\alpha_P$.

3.2 Definition of the reduction

Let $\phi \in \mathrm{DLP}^+_{\mathrm{dyn}}$ built over the atomic actions $b_1, \ldots, b_N$. We define a PDL formula ϕ' built over the atomic programs $\{b_i^g, b_i^r : 1 \leq i \leq N\}$ where "g" stands for "green", "r" stands for "red", and each pair $\langle b_i^g, b_i^r \rangle$ is interpreted as a disjoint pair of relations. ϕ' is defined from the translation function $T(\psi, \mathsf{G}, \mathsf{R})$ where G and R are finite sets of action expressions of the form

$$\psi_1?; \cdots ; \psi_n?; b; \psi'_1? \cdots ; \psi'_{n'}?$$

with $n, n' \geq 0$ and b is an atomic expression in $\{b_i^g, b_i^r : 1 \leq i \leq N\}$. G is a finite set of action expressions whose union is interpreted as the current green part of the model and R is a finite set of action expressions whose union is interpreted as its current red part. The sets R and G are then updated in the recursive calls when `revoke` and `grant` formulae are translated.

The formula ϕ' is defined as the PDL formula $T(\phi, \mathsf{G}_0, \mathsf{R}_0)$ where

$$\mathsf{G}_0 = \{b_i^g : 1 \leq i \leq N\}, \quad \mathsf{R}_0 = \{b_i^r : 1 \leq i \leq N\}.$$

Along the translation process, the union of the set G denoted by $(\bigcup_{\alpha \in \mathsf{G}} \alpha)$ [resp. R denoted by $(\bigcup_{\alpha \in \mathsf{R}} \alpha)$] is interpreted as the restriction of P [resp. $-P$] to $\bigcup_{1 \leq i \leq N}(R_{b_i^g} \cup R_{b_i^r})$. Hence, $\mathsf{G} \cup \mathsf{R}$ is always interpreted precisely to $\bigcup_{1 \leq i \leq N}(R_{b_i^g} \cup R_{b_i^r})$. Observe that if α_P is equal to $(\bigcup_{\alpha \in \mathsf{G}} \alpha)$, then $(b_i^g \cup b_i^r) \cap \alpha_P$ is equivalent to

$$\bigcup \{\psi_1?; \cdots ; \psi_n?; b_i^\bullet; \psi'_1? \cdots ; \psi'_{n'}? \in \mathsf{G} : \bullet \in \{g, r\}\}.$$

A similar observation holds for $(b_i^g \cup b_i^r) \cap -\alpha_P$. We provide below the definition of $T(\cdot)$:

* $T(p, \mathsf{G}, \mathsf{R}) = p$ and T is homomorphic for the Boolean operators $\neg$, $\wedge$.
* $T([\alpha]\psi, \mathsf{G}, \mathsf{R}) = [T(\alpha, \mathsf{G}, \mathsf{R})]T(\psi, \mathsf{G}, \mathsf{R})$ where T on actions is defined as follows:
  - $T(b_i, \mathsf{G}, \mathsf{R}) = b_i^g \cup b_i^r$,
  - T is homomorphic for the operators ;, ?, $\cup$ and *.
* $T(\mathtt{perm}(\alpha)\psi, \mathsf{G}, \mathsf{R}) = \langle t^{\forall g}(\alpha, \mathsf{G}, \mathsf{R})\rangle T(\psi, \mathsf{G}, \mathsf{R})$ where $t^{\forall g}$ is defined as follows:
  - $t^{\forall g}$ is homomorphic for the operators ;, $\cup$ and *,
  - $t^{\forall g}(b_i, \mathsf{G}, \mathsf{R}) = \bigcup\{\psi_1?; \cdots ; \psi_n?; b_i^\bullet; \psi_1'?; \cdots ; \psi_{n'}'? \in \mathsf{G} : \bullet \in \{g, r\}\}$. Since we enforce that b and $b^g \cup b^r$ have the same interpretation, for every $\alpha \in \mathsf{G}$ of the form $\psi_1?; \cdots ; \psi_n?; b_i^\bullet; \psi_1'?; \cdots ; \psi_{n'}'?$, we have $R_\alpha \subseteq R_{b_i}$. Moreover, for every $\langle s, s'\rangle \in R_\alpha$, $\langle s, s'\rangle$ is green with respect to the current value of the policy set. Indeed, $\mathtt{perm}(\beta)\psi$ holds true when there is a P-green path in $\mathrm{L}(\beta)$ from the current state that leads to a state satisfying ψ.
  - $t^{\forall g}(\psi?, \mathsf{G}, \mathsf{R}) = T(\psi, \mathsf{G}, \mathsf{R})?$.
* $T(\mathtt{freeperm}(\alpha)\psi, \mathsf{G}, \mathsf{R}) = \neg\langle t^{\exists r}(\alpha, \mathsf{G}, \mathsf{R})\rangle T(\psi, \mathsf{G}, \mathsf{R})$ where $t^{\exists r}$ is defined as follows:
  - $t^{\exists r}(\psi?, \mathsf{G}, \mathsf{R}) = \bot?$,
  - $t^{\exists r}(\alpha_1 \cup \alpha_2, \mathsf{G}, \mathsf{R}) = t^{\exists r}(\alpha_1, \mathsf{G}, \mathsf{R}) \cup t^{\exists r}(\alpha_2, \mathsf{G}, \mathsf{R})$,
  - $t^{\exists r}(\alpha_1; \alpha_2, \mathsf{G}, \mathsf{R}) = (t^{\exists r}(\alpha_1, \mathsf{G}, \mathsf{R}); T(\alpha_2, \mathsf{G}, \mathsf{R})) \cup (T(\alpha_1, \mathsf{G}, \mathsf{R}); t^{\exists r}(\alpha_2, \mathsf{G}, \mathsf{R}))$,
  - $t^{\exists r}(\alpha^*, \mathsf{G}, \mathsf{R}) = T(\alpha^*, \mathsf{G}, \mathsf{R}); t^{\exists r}(\alpha, \mathsf{G}, \mathsf{R}); T(\alpha^*, \mathsf{G}, \mathsf{R})$,
  - $t^{\exists r}(b_i, \mathsf{G}, \mathsf{R}) = \bigcup\{\psi_1?; \cdots ; \psi_n?; b_i^\bullet; \psi_1'? \cdots ; \psi_{n'}'? \in \mathsf{R} : \bullet \in \{g, r\}\}$. For every $\alpha \in \mathsf{R}$ of the form $\psi_1?; \cdots ; \psi_n?; b_i^\bullet; \psi_1'?; \cdots ; \psi_{n'}'?$, we have $R_\alpha \subseteq R_{b_i}$. Moreover, for every $\langle s, s'\rangle \in R_\alpha$, $\langle s, s'\rangle$ is red with respect to the current value of the policy set. $t^{\exists r}$ is based on the homonymic map defined in Section 2.3.2.
* $T(\mathtt{grant}(\psi_1, \psi_2)\ \psi, \mathsf{G}, \mathsf{R}) = T(\psi, \mathsf{G}', \mathsf{R}')$ with
  - $\mathsf{G}' = \mathsf{G} \cup \{T(\psi_1, \mathsf{G}, \mathsf{R})?; b_i^g; T(\psi_2, \mathsf{G}, \mathsf{R})? : 1 \leq i \leq N\} \cup \{T(\psi_1, \mathsf{G}, \mathsf{R})?; b_i^r; T(\psi_2, \mathsf{G}, \mathsf{R})? : 1 \leq i \leq N\}$ (see Lemma 3.2(III)),
  - $\mathsf{R}' = \{(\neg T(\psi_1, \mathsf{G}, \mathsf{R}))?; \alpha : \alpha \in \mathsf{R}\} \cup \{\alpha; (\neg T(\psi_2, \mathsf{G}, \mathsf{R}))? : \alpha \in \mathsf{R}\}$ (see Lemma 3.2(IV)).

$*$ $T(\texttt{revoke}(\psi_1,\psi_2)\psi,\mathsf{G},\mathsf{R}) = T(\psi,\mathsf{G}',\mathsf{R}')$ with

- $\mathsf{G}' = \{\neg T(\psi_1,\mathsf{G},\mathsf{R})?;\alpha : \alpha \in \mathsf{G}\} \cup \{\alpha;\neg T(\psi_2,\mathsf{G},\mathsf{R})? : \alpha \in \mathsf{G}\}$ (see Lemma 3.2(IV)).
- $\mathsf{R}' = \mathsf{R} \cup \{T(\psi_1,\mathsf{G},\mathsf{R})?;b_i^g;T(\psi_2,\mathsf{G},\mathsf{R})? : 1 \leq i \leq N\} \cup$ $\{T(\psi_1,\mathsf{G},\mathsf{R})?;a_i^r;T(\psi_2,\mathsf{G},\mathsf{R})? : 1 \leq i \leq N\}$ (see Lemma 3.2(III)).

There is a perfect symmetry between the definition of `revoke` and `grant` by swapping R with G, and R' with G'.

3.3 Correctness

Before stating the main soundness lemma, we need to establish Lemma 3.3 below. Although, disjointness is not a modally definable property, satisfiability is not sensitive to this additional assumption.

Lemma 3.3. A formula ϕ is $\mathrm{DLP}^+_{\mathrm{dyn}}$ satisfiable iff ϕ is $\mathrm{DLP}^+_{\mathrm{dyn}}$ satisfiable in a model such that all the relations for the atomic actions are disjoint.

Proof: Suppose that ϕ is $\mathrm{DLP}^+_{\mathrm{dyn}}$ satisfiable and $b_1,\ldots,b_N$ are the atomic actions occurring in ϕ. There is a $\mathrm{DLP}^+_{\mathrm{dyn}}$ model $\mathcal{M} = \langle W,(R_a)_{a\in\Pi_0},V,P\rangle$ and $s \in W$ such that $\mathcal{M},s \models \phi$. Let $\mathcal{M}' = \langle W',(R'_a)_{a\in\Pi_0},V',P'\rangle$ be the $\mathrm{DLP}^+_{\mathrm{dyn}}$ model defined by unfolding $\mathcal{M}$:

$*$ W' is the set of sequences of the form $s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n$ in $\mathcal{M}$ with $s_0 = s$ and for every $j \in \{1,\ldots,n\}$, $s_{j-1} \stackrel{b_{i_j}}{\to} s_j$ in $\mathcal{M}$,

$*$ $s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \; R'_{b_i} \; s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \stackrel{b_i}{\to} s_{n+1}$ for every $i \in \{1,\ldots,N\}$,

$*$ $V'(s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n) \stackrel{\text{def}}{=} V(s_n)$,

$*$ $s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \; P' \; t_0 \stackrel{b_{j_1}}{\to} t_1 \stackrel{b_{j_2}}{\to} \ldots \stackrel{b_{j_{n'}}}{\to} t_{n'} \stackrel{\text{def}}{\Leftrightarrow} \langle s_n,t_{n'}\rangle \in P$. P' is denoted by $\mathrm{unfolding}(P)$.

By structural induction, one can show that for all subformulae ψ of ϕ, for all $s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \in W'$, $\mathcal{M}',s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \models \psi$ iff $\mathcal{M},s_n \models \psi$. This lemma is easy to show for the PDL fragment of $\mathrm{DLP}^+_{\mathrm{dyn}}$ since $\mathcal{M}$ and $\mathcal{M}'$ are bisimilar [JW96, Theorem 11] and [vB98]. By way of example, we treat below the subformulae of the form $\texttt{freeperm}(\alpha)\;\psi'$ and $\texttt{grant}(\psi_1,\psi_2)\;\psi'$.
Case 1: $\psi = \texttt{freeperm}(\alpha)\;\psi'$.

Suppose that not $\mathcal{M}, s_n \models \psi$. By definition of $\models$, there is a path $t_0 \stackrel{A_0}{\to} t_1 \stackrel{A_1}{\to} \cdots \stackrel{A_{n'-1}}{\longrightarrow} t_{n'}$ in $\mathcal{M}$ such that

- $A_0 A_1 \cdots A_{n'-1} \in \mathrm{L}(\alpha)$ and $t_0 = s_n$,
- for some $i \in \{0, \ldots, n'-1\}$, $A_i \in \Pi_0$ and $\langle t_i, t_{i+1} \rangle \notin P$,
- $\mathcal{M}, t_{n'} \models \psi'$.

We write $A_{j_1}, \ldots, A_{j_l}$ to denote the elements of $A_0 A_1 \cdots A_{n-1}$ that belongs to Π_0. By using (IH) (for the $A_i \notin \Pi_0$) and the definition of R' and P', there is a path

$$(s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n) \stackrel{A_0}{\to} \ldots \stackrel{A_{n'-1}}{\longrightarrow} (s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \stackrel{b_{j_1}}{\to} t_{j_{1+1}} \ldots \stackrel{b_{j_l}}{\to} t_{j_{l+1}})$$

- $A_0 A_1 \cdots A_{n'-1} \in \mathrm{L}(\alpha)$,
- for some $i \in \{1, \ldots, l\}$, $\langle T_i, T_{i+1} \rangle \notin P$, with
  - $T_i = s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \stackrel{A_{j_1}}{\longrightarrow} t_{j_{1+1}} \ldots \stackrel{A_{j_i}}{\longrightarrow} t_{j_{i+1}}$,
  - $T_{i+1} = s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \stackrel{A_{j_1}}{\longrightarrow} t_{j_{1+1}} \ldots \stackrel{A_{j_{i+1}}}{\longrightarrow} t_{j_{i+2}}$,
- $\mathcal{M}, T_{n'} \models \psi'$ with

$$T_{n'} = s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \stackrel{A_{j_1}}{\longrightarrow} t_{j_{1+1}} \ldots \stackrel{A_{j_l}}{\longrightarrow} t_{j_{l+1}}.$$

Hence, not $\mathcal{M}', s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \models \psi$. In a similar fashion we can show that if $\mathcal{M}, s_n \models \psi$ then $\mathcal{M}', s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \models \psi$.
Case 2: $\psi = \mathtt{grant}(\psi_1, \psi_2)\ \psi'$.
By definition of $\models$, ($\star$) $\mathcal{M}, s_n \models \mathtt{grant}(\psi_1, \psi_2)\ \psi'$ iff $\langle W, (R_a)_{a \in \Pi_0}, V, P \cup P_{\mathcal{M}}^{\psi_1, \psi_2} \rangle, s_n \models \psi'$. By (IH), ($\star$) iff $\langle W', (R'_a)_{a \in \Pi_0}, V', P' \cup \mathrm{unfolding}(P'') \rangle, s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \models \psi'$ iff $\langle W', (R'_a)_{a \in \Pi_0}, V', P' \rangle, s_0 \stackrel{b_{i_1}}{\to} s_1 \stackrel{b_{i_2}}{\to} \ldots \stackrel{b_{i_n}}{\to} s_n \models \mathtt{grant}(\psi_1, \psi_2)\ \psi'$. □

Following the proof of Lemma 3.3, one can show that $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ is closed under bisimulation and therefore it will not be surprising that $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ can be encoded into the modal μ-calculus [JW96, Theorem 11], more precisely into its PDL fragment.

Lemma 3.4. ϕ is $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ satisfiable iff $T(\phi, \mathsf{G}_0, \mathsf{R}_0)$ is PDL satisfiable.

Proof: We show by induction on the structure of subformulae ψ of ϕ that for all $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ models $\mathcal{M} = \langle W, (R_a)_{a\in\Pi_0}, V, P\rangle$ and for all $s \in W$, $\mathcal{M}, s \models \psi$ iff $\langle W, (R'_a)_{a\in\Pi_0}, V\rangle, s \models T(\psi, \mathsf{G}, \mathsf{R})$ given that

(I) for every $1 \leq i \leq N$, $\{R'_{b_i^g}, R'_{b_i^r}\}$ is a partition of R_{b_i},

(II) $R_{b_1}, \ldots, R_{b_N}$ are pairwise disjoint,

(III) $\{R'_{(\bigcup_{\alpha\in\mathsf{G}}\alpha)}, R'_{(\bigcup_{\alpha\in\mathsf{R}}\alpha)}\}$ is a partition of $R_{b_1} \cup \cdots \cup R_{b_N}$,

(IV) $P = R'_{(\bigcup_{\alpha\in\mathsf{G}}\alpha)}$.

Let us first check that it is enough to prove this result. Suppose that ϕ is $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ satisfiable. Let $\mathcal{M} = \langle W, (R_a)_{a\in\Pi_0}, V, P\rangle$ be a model and $s \in W$ such that $\mathcal{M}, s \models \phi$. By Lemma 3.3 and Lemma 3.1, we can assume that $R_{b_1}, \ldots, R_{b_N}$ are pairwise disjoint and $P \subseteq \bigcup_{1\leq i\leq N} R_{b_i}$. Let $\mathcal{M}' = \langle W, (R'_a)_{a\in\Pi_0}, V\rangle$ be the PDL model such that for every $i \in \{1, \ldots, N\}$, $R'_{b_i^g} = R_{b_i} \cap P$ and $R'_{b_i^r} = R_{b_i} \cap -P$. It is easy to check that (I)-(IV) hold true with $\mathsf{G} = \mathsf{G}_0$ and $\mathsf{R} = \mathsf{R}_0$. Hence, $\mathcal{M}', s \models T(\psi, \mathsf{G}_0, \mathsf{R}_0)$.
Similarly, suppose that $T(\psi, \mathsf{G}_0, \mathsf{R}_0)$ is PDL satisfiable. So there are a model $\mathcal{M} = \langle W, (R_a)_{a\in\Pi_0}, V\rangle$ and $s \in W$ such that $\mathcal{M}, s \models \phi$. By Lemma 3.3, we can also assume that $R_{b_1^g}, R_{b_1^r}, \ldots, R_{b_N^g}, R_{b_N^r}$ are pairwise disjoint. Let $\mathcal{M}' = \langle W, (R'_a)_{a\in\Pi_0}, V, P'\rangle$ be the $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ model such that for every $i \in \{1, \ldots, N\}$, $R'_{b_i} = R_{b_i^g} \cup R_{b_i^r}$ and $P = R_{b_1^g} \cup \cdots \cup R_{b_N^g}$. It is easy to check that (I)-(IV) hold true with $\mathsf{G} = \mathsf{G}_0$ and $\mathsf{R} = \mathsf{R}_0$. Hence, $\mathcal{M}', s \models \psi$.

The proof by induction is immediate for the cases ψ is atomic and when the outermost operator is either $\neg$ or $\wedge$.
Case 1: $\psi = [\alpha]\psi'$.
By definition of $\models$, $\mathcal{M}, s \models [\alpha]\psi'$ iff $(\star)$ for all paths of the form $s_0 \xrightarrow{A_0} s_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} s_n$ in $\mathcal{M}$ with $A_0A_1\cdots A_{n-1} \in \mathrm{L}(\alpha)$ and $s_0 = s$, we have $\mathcal{M}, s_n \models \psi'$. $t \xrightarrow{b_i} t'$ in $\mathcal{M}$ iff either $t \xrightarrow{b_i^g} t'$ or $t \xrightarrow{b_i^r} t'$ in $\mathcal{M}'$ and by (IH) for every subformula ψ'' occurring in ψ, $t \xrightarrow{\psi''?} t'$ in $\mathcal{M}$ iff $t \xrightarrow{T(\psi'',\mathsf{G},\mathsf{R})?} t'$ in $\mathcal{M}'$. Hence by using the definition of T on program expressions and the (IH) once more, we obtain

$(\star)$ iff for all paths of the form $s_0 \xrightarrow{A_0} s_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} s_n$ in $\mathcal{M}'$ with
$A_0A_1\cdots A_{n-1} \in \mathrm{L}(T(\alpha, \mathsf{G}, \mathsf{R}))$ and $s_0 = s$,
we have $\mathcal{M}', s_n \models T(\psi', \mathsf{G}, \mathsf{R})$,
iff $\mathcal{M}', s \models [T(\alpha, \mathsf{G}, \mathsf{R})]T(\psi', \mathsf{G}, \mathsf{R})$,
iff $\mathcal{M}', s \models T([\alpha]\psi, \mathsf{G}, \mathsf{R})$.

Case 2: $\psi = \mathtt{perm}(\alpha)\psi'$.
By definition of $\models$, $\mathcal{M}, s \models \mathtt{perm}(\alpha)\psi'$ iff ($\star\star$) there is a path of the form $s_0 \stackrel{A_0}{\rightarrow} s_1 \stackrel{A_1}{\rightarrow} \cdots \stackrel{A_{n-1}}{\longrightarrow} s_n$ in $\mathcal{M}$ such that

- $A_0 A_1 \cdots A_{n-1} \in \mathrm{L}(\alpha)$, $s_0 = s$, and
- for every $i \in \{1, \ldots, N\}$, $A_i \in \Pi_0$ implies $\langle s_i, s_{i+1} \rangle \in P$,
- $\mathcal{M}, s_n \models \psi'$.

Observe that $s_i \stackrel{A_i}{\rightarrow} s_{i+1}$ with $A_i = b_j$ in $\mathcal{M}$ and $\langle s_i, s_{i+1} \rangle \in P$ is equivalent to either

- $s_i \stackrel{b_j^g}{\rightarrow} s_{i+1}$ in $\mathcal{M}'$ and
- there is $\psi_1?; \cdots; \psi_n?; b_j^g; \psi_1'? \cdots; \psi_{n'}'? \in \mathsf{G}$ such that $\mathcal{M}', s_i \models \psi_1 \wedge \cdots \wedge \psi_n$ and $\mathcal{M}', s_{i+1} \models \psi_1' \wedge \cdots \wedge \psi_{n'}'$,

or

- $s_i \stackrel{b_j^r}{\rightarrow} s_{i+1}$ in $\mathcal{M}'$ and
- there is $\psi_1?; \cdots; \psi_n?; b_j^r; \psi_1'? \cdots; \psi_{n'}'? \in \mathsf{G}$ such that $\mathcal{M}', s_i \models \psi_1 \wedge \cdots \wedge \psi_n$ and $\mathcal{M}', s_{i+1} \models \psi_1' \wedge \cdots \wedge \psi_{n'}'$.

This equivalence holds true because of the satisfaction of (I) and (II). By (IH) for every subformula ψ'' occurring in ψ, we still have $t \stackrel{\psi''?}{\longrightarrow} t'$ in $\mathcal{M}$ iff $t \stackrel{T(\psi'', \mathsf{G}, \mathsf{R})?}{\longrightarrow} t'$ in $\mathcal{M}'$. Hence, ($\star\star$) is equivalent to: there is a path of the form $s_0 \stackrel{A_0}{\rightarrow} s_1 \stackrel{A_1}{\rightarrow} \cdots \stackrel{A_{n-1}}{\longrightarrow} s_n$ in $\mathcal{M}'$ with

- $A_0 A_1 \cdots A_{n-1} \in \mathrm{L}(T(\alpha, \mathsf{G}, \mathsf{R}))$, $s_0 = s$, and
- for every $i \in \{1, \ldots, N\}$,
  $A_i = b_j$ for some j implies there is $\psi_1?; \cdots; \psi_n?; b_j^\bullet; \psi_1'? \cdots; \psi_{n'}'? \in \mathsf{G}$,
- $\mathcal{M}', s_n \models T(\psi', \mathsf{G}, \mathsf{R})$.

It is then not difficult to show that ($\star\star$) is equivalent to there is a path of the form $s_0 \stackrel{A_0}{\rightarrow} s_1 \stackrel{A_1}{\rightarrow} \cdots \stackrel{A_{n-1}}{\longrightarrow} s_n$ in $\mathcal{M}'$ with $A_0 A_1 \cdots A_{n-1} \in \mathrm{L}(t^{\forall g}(\alpha, \mathsf{G}, \mathsf{R}))$, $s_0 = s$, and $\mathcal{M}', s_n \models T(\psi', \mathsf{G}, \mathsf{R})$. So $\mathcal{M}', s \models \langle t^{\forall g}(\alpha, \mathsf{G}, \mathsf{R}) \rangle T(\psi, \mathsf{G}, \mathsf{R})$.
Case 3: $\psi = \mathtt{grant}(\phi_1, \phi_2)\psi'$.

$$\begin{aligned} \mathcal{M}, s \models \mathtt{grant}(\phi_1, \phi_2)\, \psi' \text{ iff } & \langle W, (R_a)_{a \in \Pi_0}, V, P \cup P_{\mathcal{M}}^{\phi_1, \phi_2} \rangle, s \models \psi' \\ \text{iff } & \langle W, (R_a)_{a \in \Pi_0}, V, P \cup P_{\mathcal{M}}^{\phi_1, \phi_2} \cap R \rangle, s \models \psi' \\ & \text{with } R = (\textstyle\bigcup_{1 \leq i \leq N} R_{b_i}) \end{aligned}$$

iff $\langle W, (R_a)_{a\in\Pi_0}, V, P \cup P_1''\rangle, s \models \psi'$ with
$P_1'' = \{\langle t, t'\rangle \in (\bigcup_{1\le i\le N} R_{b_i}) : \mathcal{M}', t \models T(\psi_1, \mathsf{G}, \mathsf{R})$ and
$\mathcal{M}', t' \models T(\psi_2, \mathsf{G}, \mathsf{R})\}$

iff $\langle W, (R_a)_{a\in\Pi_0}, V, P \cup P_2''\rangle, s \models \psi'$ with
$P_2'' = \{\langle t, t'\rangle \in (\bigcup_{1\le i\le N} R'_{b_i^g} \cup R'_{b_i^r}) : \mathcal{M}', t \models T(\psi_1, \mathsf{G}, \mathsf{R}),$ and
$\mathcal{M}', t' \models T(\psi_2, \mathsf{G}, \mathsf{R})\}$

iff $\langle W, (R_a)_{a\in\Pi_0}, V, P \cup P_3''\rangle, s \models \psi'$ with
$P_3'' = \bigcup_{1\le i\le N} R'_{T(\psi_1,\mathsf{G},\mathsf{R})?;b_i^g;T(\psi_2,\mathsf{G},\mathsf{R})?} \cup R'_{T(\psi_1,\mathsf{G},\mathsf{R})?;b_i^r;T(\psi_2,\mathsf{G},\mathsf{R})?}$

iff $\mathcal{M}, s \models T(\psi', \mathsf{G}_{new}, \mathsf{R}_{new})$ with
G_{new} is the union of the two sets below:

$\mathsf{G} \cup \{T(\psi_1, \mathsf{G}, \mathsf{R})?; b_i^g; T(\psi_2, \mathsf{G}, \mathsf{R})? : 1 \le i \le N\}$

$\{T(\psi_1, \mathsf{G}, \mathsf{R})?; b_i^r; T(\psi_2, \mathsf{G}, \mathsf{R})? : 1 \le i \le N\}$

$\mathsf{R}_{new} = \{(\neg T(\psi_1, \mathsf{G}, \mathsf{R}))?; \alpha : \alpha \in \mathsf{R}\} \cup \{\alpha; (\neg T(\psi_2, \mathsf{G}, \mathsf{R}))? : \alpha \in \mathsf{R}\}.$

Observe that

* $\{R'_{(\bigcup_{\alpha\in\mathsf{G}_{new}}\alpha)}, R'_{(\bigcup_{\alpha\in\mathsf{R}_{new}}\alpha)}\}$ is a partition of $R_{b_1} \cup \cdots \cup R_{b_N}$,
* $P_3'' = R'_{(\bigcup_{\alpha\in\mathsf{G}_{new}}\alpha)}$.

The cases with `freeperm` and `revoke` are analogous. □

Corollary 3.5. $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ is decidable.

3.4 Complexity upper bounds

The reduction from $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ into PDL we have defined increases exponentially the size of formulae. More precisely, $|T(\phi)|$ is in $\mathcal{O}(|\phi| \times 2^{cd(\phi)^2})$ and computing $T(\phi)$ requires also time in $\mathcal{O}(|\phi| \times 2^{cd(\phi)^2})$. Therefore $\mathrm{DLP}^{+}_{\mathrm{dyn}}$-satisfiability is in 2EXPTIME, which is a bit worst than the NEXPTIME upper bound for $\mathrm{DLP}_{\mathrm{dyn}}$ from [PW04]. However, our translation allows us to reuse all the theorem proving machinery for PDL [VW86, Tuo90].

It is reasonable to consider fragments of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ with a fixed change depth. For such fragments, our translation provides an optimal complexity bound:

Corollary 3.6. For every fixed $k \ge 0$, the $\mathrm{DLP}^{+}_{\mathrm{dyn}}$-satisfiability problem restricted to formulae of change depth at most k is EXPTIME-complete.

Since the case $k = 0$ corresponds to DLP [vdM96] we obtain the following consequence.

Corollary 3.7. DLP-satisfiability is EXPTIME-complete.

Hence, we improve the NEXPTIME upper bound established in [vdM96, PW04]. EXPTIME-hardness simply holds because the PDL fragment of DLP is known to be EXPTIME-hard. It is conjectured in [vdM96, PW04] that DLP is in EXPTIME by adapting Pratt's proof [Pra79]. Obviously, such an adaption is not completely immediate, but our reduction is a strong witness of the feasibility of such an approach. However, a by-product of our reduction is precisely the EXPTIME upper bound of DLP satisfiability.

The reduction from $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ into PDL can be viewed as a logarithmic-space reduction if the program terms in PDL are encoded as DAGs. Unfortunately, PDL with program expressions encoded as DAGs is EXPSPACE-hard (because the equivalence problem for regular expressions encoded as DAGs is already EXPSPACE-hard [MS72]) and it is in 2EXPTIME. Still, there is some hope that $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ is in EXPTIME. At least two possibilities to prove this bound: to invent a translation into PDL with automata, see e.g. [HKT00], or to show that PDL augmented with the new operator on programs $\oplus(\alpha, \psi_1, \psi_2)$ (with semantics $(\psi_1?;\alpha) \cup (\alpha;\psi_2?)$) is in EXPTIME.

4 A variant with a violation constant

In [vdM96], the introduction of the logic DLP was motivated by the need to replace the concept of permitted states of affairs from [Mey88] by the concept of permitted actions. In [PW04], the logic DLP is extended, providing the logic $\mathrm{DLP}_{\mathrm{dyn}}$, in such a way that in the logical language the policy set can be updated dynamically via the operators `grant` and `revoke`. We recall that a policy set in [vdM96] is simply a set of transitions, a subset of $W \times W$ when the set of states is W whereas a policy set in [Mey88] is rather a subset of W (to distinguish the good states from the bad ones). The way to handle a policy set in [Mey88] is quite simple: a distinguished propositional variable is introduced (say vc) and it represents the violation constant, i.e. no action can lead to a state satisfying vc. From a formal viewpoint, having a distinguished propositional variable in PDL does not cause any difficulty whereas developments made in [vdM96, PW04] are evidence that considering a policy set as a set of transitions is technically much harder.

We define below a variant of Meyer's logic [Mey88] with the ability to update dynamically the interpretation of the violation constant vc as $\mathrm{DLP}_{\mathrm{dyn}}$ [PW04] is the dynamic counterpart of DLP [vdM96]. We call this logic $\mathrm{PDL}_{\mathrm{dyn}}$. We shall show that $\mathrm{PDL}_{\mathrm{dyn}}$ admits a simple translation into PDL, however it is not clear that this newly introduced logic $\mathrm{PDL}_{\mathrm{dyn}}$ is so interesting from the deontic viewpoint.

Nevertheless, we include the treatment of such a logic herein in order to emphasize the difference of difficulty compared with the treatment for $\mathrm{DLP}^{+}_{\mathrm{dyn}}$.

Given $\Pi_0 = \{a_i : i \geq 1\}$ and $\mathrm{PROP} = \{p_i : i \geq 1\}$, we define the set Π of action expressions and the set FOR of formulae for $\mathrm{PDL}_{\mathrm{dyn}}$ inductively as follows:

$$\Pi \ni \alpha, \beta ::= a_i \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid \phi?$$

$$\mathrm{FOR} \ni \phi, \psi ::= vc \mid p_i \mid \phi \wedge \psi \mid \neg\phi \mid [\alpha]\phi \mid \mathtt{grant}(\psi)\phi \mid \mathtt{revoke}(\psi)\phi$$

where vc is a distinguished propositional variable not present in PROP. A $\mathrm{PDL}_{\mathrm{dyn}}$-model $\mathcal{M}$ is a structure of the form $\langle W, (R_a)_{a\in\Pi_0}, V, VC\rangle$ where

- W is a non-empty set of states,
- $(R_a)_{a\in\Pi_0}$ is a family of binary relations over W,
- $V : W \to \mathcal{P}(\mathrm{PROP})$,
- $VC \subseteq W$ is the set of forbidden states.

A $\mathrm{PDL}_{\mathrm{dyn}}$-model is a PDL-model over the atomic formulae in $\mathrm{PROP} \cup \{vc\}$. We say that the formula ϕ is satisfied in the model $\mathcal{M}$ by the state s if the following conditions are satisfied (obvious clauses are omitted):

- $\mathcal{M}, s \models vc \overset{\text{def}}{\Leftrightarrow} s \in VC$,
- $\mathcal{M}, s \models \mathtt{grant}(\psi)\ \phi \overset{\text{def}}{\Leftrightarrow} \mathcal{M}', s \models \phi$ where $\mathcal{M}'$ is obtained from $\mathcal{M}$ by replacing VC by $VC \cup \{s \in W : \mathcal{M}, s \models \psi\}$,
- $\mathcal{M}, s \models \mathtt{revoke}(\psi)\ \phi \overset{\text{def}}{\Leftrightarrow} \mathcal{M}', s \models \phi$ where $\mathcal{M}'$ is obtained from $\mathcal{M}$ by replacing VC by $VC \setminus \{s \in W : \mathcal{M}, s \models \psi\}$.

Hence, in the $\mathrm{PDL}_{\mathrm{dyn}}$ semantics, the operator `grant` enriches VC and the operator `revoke` impoverishes VC. Satisfiability and other similar notions are defined in the obvious way. We define below a two-place translation $t(\cdot,\cdot)$ from $\mathrm{PDL}_{\mathrm{dyn}}$ formulae into PDL formulae (augmented with the propositional variable vc) such that $t(\phi, \psi)$ is the translation of ϕ in PDL when vc is interpreted by $\{s \in W : \mathcal{M}, s \models \psi\}$.

- $t(vc, \psi) = \psi$, $t(p, \psi) = p$,
- t is homomorphic for $\neg$ and $\wedge$,
- $t([\alpha]\phi, \psi) = [t(\alpha, \psi)]t(\phi, \psi)$,
- $t(\mathtt{grant}(\phi')\phi, \psi) = t(\phi, \psi \vee t(\phi', \psi))$,
- $t(\mathtt{revoke}(\phi')\phi, \psi) = t(\phi, \psi \wedge \neg t(\phi', \psi))$,

* $t(a, \psi) = a$ and t is homomorphic for $\cup$, ; and *,
* $t(\phi?, \psi) = t(\phi, \psi)?$.

It is not difficult to show that t is a polynomial-time reduction. If ψ is a PDL formula then $t(\phi, \psi)$ is also a PDL formula built over propositional variables in $\mathrm{PROP} \cup \{vc\}$. The major result of this section is the following:

Lemma 4.1. ϕ is $\mathrm{PDL}_{\mathrm{dyn}}$ satisfiable iff $t(\phi, vc)$ is PDL satisfiable.

Proof: By induction on the size of ϕ we show the following property:

Let $\mathcal{M}_0 = \langle W, (R_a)_{a\in\Pi_0}, V, VC_0\rangle$ and $\mathcal{M} = \langle W, (R_a)_{a\in\Pi_0}, V, VC\rangle$ be $\mathrm{PDL}_{\mathrm{dyn}}$-models such that $VC = \{s \in W : \mathcal{M}_0, s \models \psi\}$ for some PDL formula ψ ($\mathcal{M}$ variant of $\mathcal{M}_0$). Then, for every $s \in W$, $\mathcal{M}, s \models \phi$ iff $\mathcal{M}_0, s \models t(\phi, \psi)$.

Before showing this property, let us prove that it is enough for our proof. If there exist a $\mathrm{PDL}_{\mathrm{dyn}}$-model $\mathcal{M}_0 = \langle W, (R_a)_{a\in\Pi_0}, V, VC_0\rangle$ and $s \in W$ such that $\mathcal{M}_0, s \models \phi$, then $\mathcal{M}_0, s \models t(\phi, vc)$ since $VC_0 = \{s' \in W : \mathcal{M}_0, s' \models vc\}$. Conversely, if there exist a PDL model $\mathcal{M}_0 = \langle W, (R_a)_{a\in\Pi_0}, V, VC_0\rangle$ and $s \in W$ such that $\mathcal{M}_0, s \models t(\phi, vc)$, then $\mathcal{M}_0, s \models \phi$ since $VC_0 = \{s' \in W : \mathcal{M}_0, s' \models vc\}$.

The base case when $\phi \in \mathrm{PROP} \cup \{vc\}$ and the cases when the outermost connective of ϕ is either $\wedge$ or $\neg$ are by an easy verification.

Let $\mathcal{M} = \langle W, (R_a)_{a\in\Pi_0}, V, VC\rangle$ be a $\mathrm{PDL}_{\mathrm{dyn}}$-model variant of $\mathcal{M}_0$ such that $VC = \{s \in W : \mathcal{M}_0, s \models \psi\}$.
Case 1: $\phi = \mathtt{grant}(\phi')\phi''$.

$\mathcal{M}, s \models \mathtt{grant}(\phi')\phi''$ iff $\mathcal{M}', s \models \phi''$ with $VC' = VC \cup \{s' : \mathcal{M}, s' \models \phi'\}$
(by definition of $\models$)
iff $\mathcal{M}', s \models \phi''$ with $VC' = \{s' \in W : \mathcal{M}_0, s' \models \psi\} \cup \{s' : \mathcal{M}_0, s' \models t(\phi', \psi)\}$
(by assumption and by (IH))
iff $\mathcal{M}_0, s \models t(\phi'', \psi \vee t(\phi', \psi))$ (by (IH)).

Case 2: $\phi = \mathtt{revoke}(\phi')\phi''$.

$\mathcal{M}, s \models \mathtt{revoke}(\phi')\phi''$ iff $\mathcal{M}', s \models \phi''$ with $VC' = VC \setminus \{s' : \mathcal{M}, s' \models \phi'\}$
(by definition of $\models$)
iff $\mathcal{M}', s \models \phi''$ with $VC' = \{s' \in W : \mathcal{M}_0, s' \models \psi\} \setminus \{s' : \mathcal{M}_0, s' \models t(\phi', \psi)\}$
(by assumption and by (IH))
iff $\mathcal{M}_0, s \models t(\phi'', \psi \wedge \neg t(\phi', \psi))$.

Case 3: $\phi = [\alpha]\phi'$.

For any subexpression $\varphi?$ in α, by (IH) for all $s' \in W$, $s' \xrightarrow{\varphi} s'$ in $\mathcal{M}$ iff $s' \xrightarrow{t(\varphi,\psi)} s'$ in $\mathcal{M}_0$. Since the map t is homomorphic for $\cup$, ; and *, then for all $s', s'' \in W$, $\langle s', s''\rangle \in R_\alpha$ in $\mathcal{M}$ iff $\langle s', s''\rangle \in R_{t(\alpha,\psi)}$ in $\mathcal{M}_0$. It is then easy to show that $\mathcal{M}, s \models [\alpha]\phi'$ iff $\mathcal{M}_0, s \models t([\alpha]\phi', \psi)$. □

Since t is a polynomial-time reduction, complexity of $\mathrm{PDL}_{\mathrm{dyn}}$ can be now easily characterized.

Corollary 4.2. Satisfiability for $\mathrm{PDL}_{\mathrm{dyn}}$ is EXPTIME-complete.

5 Concluding remarks

In this paper, we have shown that the logic $\mathrm{DLP}^{+}_{\mathrm{dyn}}$, a substantial extension of the logic $\mathrm{DLP}_{\mathrm{dyn}}$ (itself extending the logic DLP [vdM96]), can be naturally translated in standard PDL. As a consequence, we can reuse several results and techniques about PDL and we establish new complexity bounds for DLP and $\mathrm{DLP}_{\mathrm{dyn}}$.

Even though $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ has the ability to withdraw (or add) transitions in the policy set, this extension remains decidable. This is in sharp contrast with a variant of the sabotage modal logic SML which has an undecidable satisfiability problem [LR03]. This can be explained by the fact that in $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ there is no quantification over all possibilities of withdrawal of transitions/states (as in SML), but rather that one needs to specify the formulae satisfied by the extremity nodes of the transitions to be withdrawn or added. This is a situation similar to the decidability of the guarded fragment of classical logic [ANvB98] where quantification has a guarded flavor. Actually, in $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ there is no quantification on the different ways to modify policy sets.

Even if our translation into PDL sheds some new light on the deontic logics $\mathrm{DLP}_{\mathrm{dyn}}$ [PW04] and DLP [vdM96], some interesting problems remain open including

- the characterization of the precise complexity of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ and $\mathrm{DLP}_{\mathrm{dyn}}$ (between EXPTIME and 2EXPTIME),
- the decidability status of $\mathrm{DLP}^{+}_{\mathrm{dyn}}$ with the semantics in which the atomic actions are interpreted as sets of finite sequences of transitions (on the model of the original DLP semantics), see Section 2.2 for more details. Indeed, our translation technique does not seem to work for this extension.

References

[ANvB98] H. Andreka, I. Nemeti, and J. van Benthem. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.

[Bro03] J. Broersen. *Modal action logics for Reasoning about Reactive systems*. PhD thesis, Dutch Research School for Information and Knowledge Systems, 2003.

[CS91] R. Cleaveland and B. Steffen. A linear-time model-checking algorithm for the alternation-free modal mu-calculus. In K. Larsen and A. Skou, editors, *Computer-Aided Verification (CAV '91), Aalborg, Denmark*, volume 575 of *Lecture Notes in Computer Science*, pages 48–58. Springer-Verlag, 1991.

[Dan84] R. Danecki. Nondeterministic propositional dynamic logic with intersection is decidable. In A. Skowron, editor, *5th Symposium on Computation Theory, Zaborów, Poland*, volume 208 of *Lecture Notes in Computer Science*, pages 34–53. Springer-Verlag, 1984.

[Dem01] S. Demri. The complexity of regularity in grammar logics and related modal logics. *Journal of Logic and Computation*, 11(6):933–960, 2001.

[DG00] S. Demri and D. Gabbay. On modal logics characterized by models with relative accessibility relations: Part I. *Studia Logica*, 65(3):323–353, 2000.

[dGL94] G. de Giacomo and M. Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics. In *AAAI'94*, pages 205–212. AAAI Press/The MIT Press, 1994.

[FI87] M. Fischer and N. Immerman. Interpreting logics of knowledge in propositional dynamic logic with converse. *Information Processing Letters*, 25:175–181, 1987.

[FL79] M. Fischer and R. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18:194–211, 1979.

[HKT00] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.

[JW96] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In U. Montanari and V. Sassone, editors, *CONCUR'96*, volume 1119 of *Lecture Notes in Computer Science*, pages 263–277. Springer-Verlag, 1996.

[LR03] Ch. Löding and Ph. Rohde. Model checking and satisfiability for sabotage modal logic. In P. Pandya and J. Radhakrishnan, editors, *FSTTCS'03*, volume 2914 of *Lecture Notes in Computer Science*, pages 302–313. Springer, 2003.

[Lut05] C. Lutz. PDL with intersection and converse is decidable. In *CSL'05*, 2005. to appear.

[LW04] C. Lutz and D. Walther. PDL with negation of atomic programs. In D. Basin and M. Rusinowitch, editors, *IJCAR'04*, volume 3097 of *Lecture Notes in Computer Science*, pages 259–273. Springer-Verlag, 2004.

[Mey88] J.J. Ch. Meyer. 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, 1988.

[MS72] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *FOCS'72*, pages 125–129, 1972.

[Pra79] V. Pratt. Models of program logics. In *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science*, pages 115–122, 1979.

[PW04] R. Pucella and V. Weissman. Reasoning about dynamic policies. In I. Walukiewicz, editor, *FOSSACS'04, Barcelona*, volume 2987 of *Lecture Notes in Computer Science*, pages 453–467, 2004.

[Roh04] Ph. Rohde. Moving in a crumbling network: the balanced case. In J. Marcinkowski and A. Tarlecki, editors, *CSL'04*, volume 3210 of *Lecture Notes in Computer Science*, pages 310–324. Springer-Verlag, 2004.

[Sch91] K. Schild. A correspondence theory for terminological logics: preliminary report. In *IJCAI-12*, pages 466–471, 1991.

[STH04] R. Schmidt, D. Tishkovsky, and U. Hustadt. Interaction between knowledge, action and commitment within agent dynamic logic. *Studia Logica*, 78(3):381–415, 2004.

[Tuo90] H. Tuominen. Dynamic logic as a uniform framework for theorem proving in intensional logic. In M. Stickel, editor, *CADE-10*, volume 449 of *Lecture Notes in Computer Science*, pages 514–527. Springer-Verlag, 1990.

[vB98] J. van Benthem. Program constructions that are safe for bisimulation. *Studia Logica*, 60:311–330, 1998.

[vB02] J. van Benthem. An essay on sabotage and obstruction. In *Festschrift in honour of Jörg Siekmann*, LNAI, 2002. to appear.

[vdM96] R. van der Meyden. The dynamic logic of permission. *Journal of Logic and Computation*, 6(3):465–479, 1996.

[VW86] M. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Sciences*, 32:183–221, 1986.

[WM94] R. Wieringa and J.-J. Ch. Meyer. Applications of deontic logic to computer science: a concise overview. In R. Wieringa and J.-J. Ch. Meyer, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 17–40. John Wiley and Sons Ltd, 1994.