Definability and Computability for $PRSPDL$

Philippe Balbiani [1]

Institut de recherche en informatique de Toulouse
CNRS — Université de Toulouse

Tinko Tinchev [2]

Department of Mathematical Logic and Applications
Sofia University

Abstract

$PRSPDL$ is a variant of PDL with parallel composition. In the Kripke models in which $PRSPDL$-formulas are evaluated, states have an internal structure. We devote this paper to the definability issue of several classes of frames by means of the language of $PRSPDL$ and to the computability issue of $PRSPDL$-validity for various fragments of the $PRSPDL$-language and for various classes of $PRSPDL$-frames.

Keywords: Propositional dynamic logic; parallel composition; definability; computability.

1 Introduction

Propositional dynamic logic (PDL) is a non-classical logic designed for reasoning about the behaviour of programs [11,16,19]. Its syntax is based on the idea of associating with each program α of some programming language the modal operator $[\alpha]$, formulas $[\alpha]\phi$ being read "every execution of α from the present state leads to a state bearing the formula ϕ". Syntactically, PDL is a modal logic with a structure in the set of modal operators: composition $(\alpha;\beta)$ of programs α and β corresponds to the composition of the accessibility relations $R(\alpha)$ and $R(\beta)$; test $\phi?$ on formula ϕ corresponds to the partial identity relation in the subsets of the Kripke models in which the formula ϕ is true; iteration $\alpha^\star$ corresponds to the reflexive and transitive closure of $R(\alpha)$. A number

[1] Address: Institut de recherche en informatique de Toulouse, CNRS — Université de Toulouse, 118 route de Narbonne, 31062 Toulouse Cedex 9, FRANCE; balbiani@irit.fr.

Partially supported by the "French National Research Agency" (contract ANR-11-BS02-011) and the "Bulgarian National Science Fund" (contract DID02/32/2009).

[2] Address: Department of Mathematical Logic and Applications, Sofia University, Blvd James Bouchier 5, 1126 Sofia, Bulgaria; tinko@fmi.uni-sofia.bg.

Partially supported by the "Centre international de mathématiques et d'informatique" (contract ANR-11-LABX-0040-CIMI within the program ANR-11-IDEX-0002-02).

of variants have been obtained by extending or restricting the syntax or the semantics of PDL [3,4,5,9,10,15,17,21,22].

The problem with most of these variants is that the states of the Kripke models in which formulas are evaluated have no internal structure. However, in the field of non-classical logics, it seems natural to propose formalisms with which one can cope with structured data such as heaps, pointers, etc. In addition to the standard Boolean constructs, separation logics are based on the formula construct $(\cdot \circ \cdot)$ of separating conjunction, formulas $(\phi \circ \psi)$ being read "the memory model can be split into two disjoint models respectively satisfying ϕ and ψ", and the formula construct $(\cdot\!-\!\!\circ\cdot)$ of adjoint implication, formulas $(\phi\!-\!\!\circ\psi)$ being read "if the memory model is extended with a model satisfying ϕ, then the resulting model satisfies ψ" [7,8,14,25]. In order to illustrate the significance of these constructs, one may consider the set of all words on an alphabet and its associated operation of concatenation, the set of all binary trees and its associated operation of join and the set of all heaps (partially defined functions mapping locations to values) and its associated operation of union (undefined when domains overlap).

$PRSPDL$, the propositional dynamic logic with storing, recovering and parallel composition introduced by Benevides *et al.* [5], is a separation-based non-classical logic too. Benevides *et al.* [5] extend the semantics of PDL by considering Kripke models structured by means of a function $*$: the state x is the result of applying the function $*$ to the states y, z iff x can be separated in a first part y and a second part z. They extend the syntax of PDL as well by adding the program construct $(\cdot \parallel \cdot)$ of parallel composition, the storing programs s_1 and s_2 and the recovering programs r_1 and r_2. In this variant, parallel composition $(\alpha \parallel \beta)$ corresponds to the fork $R(\alpha)\nabla R(\beta)$ of $R(\alpha)$ and $R(\beta)$ defined as follows:

- whenever x and y are related via $R(\alpha)$ and z and t are related via $R(\beta)$, $x * z$ and $y * t$ are related via $R(\alpha)\nabla R(\beta)$.

About s_1 and s_2, x is related, by s_1, to the states $x * z$ and, by s_2, to the states $z * x$. As for r_1 and r_2, the states $x * z$, by r_1, and the states $z * x$, by r_2, are related to x. Hence, s_1, s_2, r_1 and r_2 enable us to view states as ordered pairs of states. The function $*$ considered in [5] has its origin in the addition of an extra binary operation of fork denoted ∇ in relation algebras [12,13].

It appears that $(\cdot \parallel \cdot)$ can be eliminated from the language of $PRSPDL$ extended with $(\cdot \cap \cdot)$. To see this, it suffices to consider the equivalence between $(\alpha \parallel \beta)$ and $((r_1; \alpha; s_1) \cap (r_2; \beta; s_2))$ in all Kripke models structured by means of a function $*$ as above. On one hand, the decidability of PDL with intersection [9] seems to indicate that $PRSPDL$-validity is decidable as well. The problem is that the language of $PRSPDL$ contains two programs, namely r_1 and r_2, interpreted in [5] by deterministic binary relations. Hence, Danecki's result cannot be directly applied. On the other hand, the undecidability of PDL with intersection and at least two program variables interpreted by deterministic binary relations [18] seems to indicate that $PRSPDL$-validity is undecidable as well. The problem is that $(\cdot \cap \cdot)$ cannot be defined in the language of $PRSPDL$. Thus, Harel's result cannot be directly applied.

Nevertheless, following the line of reasoning suggested in [18], it is possible to reduce the Σ^1_1-hard $\mathbb{N} \times \mathbb{N}$ recurring tiling problem to satisfiability of $PRSPDL$-formulas when r_1 and r_2 are interpreted by deterministic binary relations as in [5]. Hence, $PRSPDL$-validity is Π^1_1-hard. The section-by-section breakdown of this article is as follows. In Section 2, we present the syntax and semantics of $PRSPDL$. The aim of Section 3 is to investigate the definability of several classes of frames. In Section 4, we demonstrate that neither the program construct $(\cdot \parallel \cdot)$, nor the storing programs s_1 and s_2, nor the recovering programs r_1 and r_2 can be eliminated from the language of $PRSPDL$. For various fragments of the $PRSPDL$-language and for various classes of $PRSPDL$-frames, we will devote Sections 5 and 6 to the computability of $PRSPDL$-validity.

2 Syntax and semantics

2.1 Syntax

Programs and formulas are inductively defined as follows:

- $\alpha ::= a \mid (\alpha;\beta) \mid \phi? \mid \alpha^\star \mid (\alpha \parallel \beta) \mid s_1 \mid s_2 \mid r_1 \mid r_2$;
- $\phi ::= p \mid \bot \mid \neg\phi \mid (\phi \vee \psi) \mid [\alpha]\phi$;

where a ranges over a countably infinite set of program variables and p ranges over a countably infinite set of propositional variables. The other Boolean constructs for formulas are defined as usual. The modal construct $\langle\cdot\rangle\cdot$ for formulas is defined as follows:

- $\langle\alpha\rangle\phi ::= \neg[\alpha]\neg\phi$.

We will follow the standard rules for omission of the parentheses.

Example 2.1 If α, β are programs and ϕ, ψ are formulas, then $\langle\alpha \parallel \beta\rangle\phi \to \langle r_1;\alpha;s_1\rangle(\phi \wedge \psi) \vee \langle r_2;\beta;s_2\rangle(\phi \wedge \neg\psi)$ is a formula as well.

Let the level of an expression exp (either a program, or a formula), in symbols $lev(exp)$, be the number of occurrences of the program construct $(\cdot \parallel \cdot)$ of parallel composition in exp.

2.2 Frames

A frame is a 3-tuple $\mathcal{F} = (W, R, *)$ where

- W is a nonempty set of states,
- R is a function from the set of all program variables into the set of all binary relations between states,
- $*$ is a function from the set of all pairs of states into the set of all sets of states.

We will use $x, y, \dots$ for states. In $\mathcal{F}$, W is to be regarded as the set of all possible states in a computation process, R associates with each program variable a the binary relation $R(a)$ on W with $xR(a)y$ meaning "y can be reached from x by performing program variable a" and $*$ associates with each pair (y, z) of states the subset $y * z$ of W with $x \in y * z$ meaning "x can be obtained as a result of the combination of y and z". We shall say that a frame $\mathcal{F} = (W, R, *)$ is functional iff for all $x, y, z \in W$, if $xR(a)y$ and $xR(a)z$, then $y = z$ for every program variable a. We will also be interested in the following types of frames:

- $*$-distributive frames, i.e. frames $\mathcal{F} = (W, R, *)$ such that for all $x, y, z, t \in W$, $(x * y) \cap (z * t) = (x * t) \cap (z * y)$,
- $*$-separated frames, i.e. frames $\mathcal{F} = (W, R, *)$ such that for all $x, y, z, t \in W$, if $(x * y) \cap (z * t) \neq \emptyset$, then $x = z$ and $y = t$,
- $*$-deterministic frames, i.e. frames $\mathcal{F} = (W, R, *)$ such that for all $x, y, z, t \in W$, if $x \in z * t$ and $y \in z * t$, then $x = y$,
- $*$-serial frames, i.e. frames $\mathcal{F} = (W, R, *)$ such that for all $x, y \in W$, $x * y \neq \emptyset$.

Remark that every $*$-separated frame is $*$-distributive. Moreover, each frame considered in [5] is $*$-separated and $*$-deterministic. In order to illustrate the significance of these types of frames, we present the following:

Example 2.2 Let W_1 be the set of all words on an alphabet and $*_1$ be the operation of concatenation. The structure $\mathcal{F}_1 = (W_1, *_1)$ is not $*$-distributive. Nevertheless, it is $*$-deterministic and $*$-serial.

Let W_2 be the set of all binary trees and $*_2$ be the operation of join. The structure $\mathcal{F}_2 = (W_2, *_2)$ is $*$-separated, $*$-deterministic and $*$-serial.

Let W_3 be the set of all heaps (partially defined functions mapping locations to values) and $*_3$ be the operation of union (undefined when domains overlap). The structure $\mathcal{F}_3 = (W_3, *_3)$ is neither $*$-distributive, nor $*$-serial. Nevertheless, it is $*$-deterministic.

2.3 Models

A model on the frame $\mathcal{F} = (W, R, *)$ is a 4-tuple $\mathcal{M} = (W, R, *, V)$ where

- V is a valuation on $\mathcal{F}$, i.e. a function from the set of all propositional variables into the set of all sets of states.

In $\mathcal{M}$, V associates with each propositional variable p the subset $V(p)$ of W with $x \in V(p)$ meaning "propositional variable p is true at x". In a model $\mathcal{M} = (W, R, *, V)$, we inductively define the properties "y can be reached from x by performing program α" (in symbols $xR_{\mathcal{M}}(\alpha)y$) and "formula ϕ is true at x" (in symbols $x \in V_{\mathcal{M}}(\phi)$) as follows:

- $xR_{\mathcal{M}}(a)y$ iff $xR(a)y$;
- $xR_{\mathcal{M}}(\alpha; \beta)y$ iff there exists $z \in W$ such that $xR_{\mathcal{M}}(\alpha)z$ and $zR_{\mathcal{M}}(\beta)y$;
- $xR_{\mathcal{M}}(\phi?)y$ iff $x = y$ and $y \in V_{\mathcal{M}}(\phi)$;
- $xR_{\mathcal{M}}(\alpha^{\star})y$ iff there exists $n \in \mathbb{N}$ and there exists $z_0, \ldots, z_n \in W$ such that $z_0 = x$, $z_0 R_{\mathcal{M}}(\alpha)z_1$, $\ldots$, $z_{n-1}R_{\mathcal{M}}(\alpha)z_n$ and $z_n = y$;
- $xR_{\mathcal{M}}(\alpha \parallel \beta)y$ iff there exists $z, t, u, v \in W$ such that $x \in z * t$, $zR_{\mathcal{M}}(\alpha)u$,

$tR_{\mathcal{M}}(\beta)v$ and $y \in u * v$;

- $xR_{\mathcal{M}}(s_1)y$ iff there exists $z \in W$ such that $y \in x * z$;
- $xR_{\mathcal{M}}(s_2)y$ iff there exists $z \in W$ such that $y \in z * x$;
- $xR_{\mathcal{M}}(r_1)y$ iff there exists $z \in W$ such that $x \in y * z$;
- $xR_{\mathcal{M}}(r_2)y$ iff there exists $z \in W$ such that $x \in z * y$;
- $x \in V_{\mathcal{M}}(p)$ iff $x \in V(p)$;
- $x \notin V_{\mathcal{M}}(\bot)$;
- $x \in V_{\mathcal{M}}(\neg\phi)$ iff $x \notin V_{\mathcal{M}}(\phi)$;
- $x \in V_{\mathcal{M}}(\phi \vee \psi)$ iff either $x \in V_{\mathcal{M}}(\phi)$, or $x \in V_{\mathcal{M}}(\psi)$;
- $x \in V_{\mathcal{M}}([\alpha]\phi)$ iff for all $y \in W$, if $xR_{\mathcal{M}}(\alpha)y$, then $y \in V_{\mathcal{M}}(\phi)$.

As a result, $x \in V_{\mathcal{M}}(\langle\alpha\rangle\phi)$ iff there exists $y \in W$ such that $xR_{\mathcal{M}}(\alpha)y$ and $y \in V_{\mathcal{M}}(\phi)$. A formula ϕ is said to be true in the model $\mathcal{M} = (W, R, *, V)$, in symbols $\mathcal{M} \models \phi$, iff $V_{\mathcal{M}}(\phi) = W$. We shall say that a formula ϕ is satisfied in $\mathcal{M}$ iff $V_{\mathcal{M}}(\phi) \neq \emptyset$. A formula ϕ is said to be valid in the frame $\mathcal{F}$, in symbols $\mathcal{F} \models \phi$, iff for all models $\mathcal{M}$ on $\mathcal{F}$, $\mathcal{M} \models \phi$. We shall say that a formula ϕ is satisfied in $\mathcal{F}$ iff there exists a model $\mathcal{M}$ on $\mathcal{F}$ such that ϕ is satisfied in $\mathcal{M}$. A formula ϕ is said to be satisfied in a class $\mathcal{C}$ of frames iff there exists a frame $\mathcal{F}$ in $\mathcal{C}$ such that ϕ is satisfied in $\mathcal{F}$.

Example 2.3 The formula $\langle\alpha \parallel \beta\rangle\phi \rightarrow \langle r_1; \alpha; s_1\rangle(\phi \wedge \psi) \vee \langle r_2; \beta; s_2\rangle(\phi \wedge \neg\psi)$ considered in Example 2.1 is valid in every $*$-separated frame.

2.4 A decision problem

Let $\mathcal{L}$ be a fragment of the $PRSPDL$-language and $\mathcal{C}$ be a class of $PRSPDL$-frames. The set of all $\mathcal{L}$-formulas that are valid in every $\mathcal{C}$-frame will be denoted $VAL(\mathcal{L}, \mathcal{C})$. For various fragments $\mathcal{L}$ of the $PRSPDL$-language and for various classes $\mathcal{C}$ of $PRSPDL$-frames, we will devote Sections 5 and 6 of this paper to the computability of the following decision problem:

- input: an $\mathcal{L}$-formula ϕ;
- output: determine whether ϕ is valid in every $\mathcal{C}$-frame.

3 Definability

A class $\mathcal{C}$ of frames is said to be modally defined by a set Σ of formulas iff for all frames $\mathcal{F}$, $\mathcal{F}$ is in $\mathcal{C}$ iff $\mathcal{F} \models \Sigma$. We shall say that a class of frames is modally definable iff it is modally defined by a set of formulas. Obviously, the class of all functional frames is modally defined by the formulas $\langle a\rangle p \rightarrow [a]p$ for every program variable a. About the class of all $*$-distributive frames, the class of all $*$-separated frames and the class of all $*$-deterministic frames, we have the following:

Proposition 3.1 1) *The class of all $*$-distributive frames is modally defined by the formula* $\langle p? \parallel \top?\rangle\top \wedge \langle\top? \parallel q?\rangle\top \rightarrow \langle p? \parallel q?\rangle\top$.

2) *The class of all $*$-separated frames is modally defined by the formulas* $\langle p? \parallel \top?\rangle\top \to [\neg p? \parallel \top?]\bot$, $\langle \top? \parallel q?\rangle\top \to [\top? \parallel \neg q?]\bot$.

3) *The class of all $*$-deterministic frames is modally defined by the formula* $p \to [\top? \parallel \top?]p$.

Proof. We only give the proof of 1), leaving the proof of 2) and 3) to the reader. Let $\mathcal{F} = (W, R, *)$ be a frame.
Suppose $\mathcal{F}$ is $*$-distributive. If $\mathcal{F} \not\models \langle p? \parallel \top?\rangle\top \wedge \langle\top? \parallel q?\rangle\top \to \langle p? \parallel q?\rangle\top$, then there exists a model $\mathcal{M} = (W, R, *, V)$ on $\mathcal{F}$ and there exists $x \in W$ such that $x \notin V_{\mathcal{M}}(\langle p? \parallel \top?\rangle\top \wedge \langle\top? \parallel q?\rangle\top \to \langle p? \parallel q?\rangle\top)$. Hence, $x \in V_{\mathcal{M}}(\langle p? \parallel \top?\rangle\top)$, $x \in V_{\mathcal{M}}(\langle\top? \parallel q?\rangle\top)$ and $x \notin V_{\mathcal{M}}(\langle p? \parallel q?\rangle\top)$. Thus, there exists $y, z, s, t \in W$ such that $x \in y * z$, $y \in V(p)$, $x \in s * t$ and $t \in V(q)$. Therefore, $x \in (y*z) \cap (s*t)$. Since $\mathcal{F}$ is $*$-distributive, then $(y*z)\cap(s*t) = (y*t)\cap(s*z)$. Since $x \in (y * z) \cap (s * t)$, then $x \in (y * t) \cap (s * z)$. Consequently, $x \in y * t$. Since $y \in V(p)$ and $t \in V(q)$, then $x \in V_{\mathcal{M}}(\langle p? \parallel q?\rangle\top)$: a contradiction.
Suppose $\mathcal{F} \models \langle p? \parallel \top?\rangle\top \wedge \langle\top? \parallel q?\rangle\top \to \langle p? \parallel q?\rangle\top$. If $\mathcal{F}$ is not $*$-distributive, then there exists $y, z, s, t \in W$ such that $(y*z)\cap(s*t) \neq (y*t)\cap(s*z)$. Hence, there exists $x \in W$ such that either $x \in (y*z) \cap (s*t)$ and $x \notin (y*t) \cap (s*z)$, or $x \in (y * t) \cap (s * z)$ and $x \notin (y * z) \cap (s * t)$. Without loss of generality, assume $x \in (y * z) \cap (s * t)$ and $x \notin (y * t) \cap (s * z)$. Thus, $x \in y * z$, $x \in s * t$ and either $x \notin y * t$, or $x \notin s * z$. Without loss of generality, assume $x \notin y * t$. Let V be a valuation on $\mathcal{F}$ such that $V(p) = \{y\}$ and $V(q) = \{t\}$. Let $\mathcal{M} = (W, R, *, V)$. Since $\mathcal{F} \models \langle p? \parallel \top?\rangle\top \wedge \langle\top? \parallel q?\rangle\top \to \langle p? \parallel q?\rangle\top$, then $x \in V_{\mathcal{M}}(\langle p? \parallel \top?\rangle\top \wedge \langle\top? \parallel q?\rangle\top \to \langle p? \parallel q?\rangle\top)$. Therefore, if $x \in V_{\mathcal{M}}(\langle p? \parallel \top?\rangle\top)$ and $x \in V_{\mathcal{M}}(\langle\top? \parallel q?\rangle\top)$, then $x \in V_{\mathcal{M}}(\langle p? \parallel q?\rangle\top)$ Since $x \in y * z$, $V(p) = \{y\}$, $x \in s * t$ and $V(q) = \{t\}$, then $x \in V_{\mathcal{M}}(\langle p? \parallel \top?\rangle\top)$ and $x \in V_{\mathcal{M}}(\langle\top? \parallel q?\rangle\top)$. Since if $x \in V_{\mathcal{M}}(\langle p? \parallel \top?\rangle\top)$ and $x \in V_{\mathcal{M}}(\langle\top? \parallel q?\rangle\top)$, then $x \in V_{\mathcal{M}}(\langle p? \parallel q?\rangle\top)$, then $x \in V_{\mathcal{M}}(\langle p? \parallel q?\rangle\top)$. Consequently, there exists $u, v \in W$ such that $x \in u * v$, $u \in V(p)$ and $v \in V(q)$. Since $V(p) = \{y\}$ and $V(q) = \{t\}$, then $u = y$ and $v = t$. Since $x \in u * v$, then $x \in y * t$: a contradiction. □

As for the class of all $*$-serial frames, we have the following:

Proposition 3.2 *The class of all $*$-serial frames is not modally definable.*

Proof. Suppose there exists a set Σ of formulas that modally defines the class of all $*$-serial frames. Let $\mathcal{F} = (W, R, *)$ and $\mathcal{F}' = (W', R', *')$ be the frames defined as follows:

- $W = \{x_1, x_2\}$,
- R is the empty function,
- $x_1 * x_1 = \{x_1\}$, $x_2 * x_2 = \{x_2\}$ and otherwise $*$ is the empty function,
- $W' = \{x'\}$,
- R' is the empty function,
- $x' *' x' = \{x'\}$.

Obviously, $\mathcal{F}$ is not $*$-serial and $\mathcal{F}'$ is $*$-serial. Since Σ modally defines the class of all $*$-serial frames, then $\mathcal{F} \not\models \Sigma$ and $\mathcal{F}' \models \Sigma$. Hence, there exists a formula $\phi \in \Sigma$ such that $\mathcal{F} \not\models \phi$. Since $\mathcal{F}' \models \Sigma$, then $\mathcal{F}' \models \phi$. Since $\mathcal{F} \not\models \phi$, then there exists a model $\mathcal{M} = (W, R, *, V)$ on $\mathcal{F}$ such that either $x_1 \notin V_{\mathcal{M}}(\phi)$, or $x_2 \notin V_{\mathcal{M}}(\phi)$. Without loss of generality, assume $x_1 \notin V_{\mathcal{M}}(\phi)$. Let $\mathcal{M}' = (W', R', *', V')$ be the model on $\mathcal{F}'$ defined as follows:

- $V'(p) =$ if $x_1 \in V(p)$, then $\{x'\}$, else $\emptyset$ for every propositional variable p.

Since $\mathcal{F}' \models \phi$, then $x' \in V_{\mathcal{M}'}(\phi)$.

Claim 3.3 *Let α be a program and ψ be a formula from the language of $PRSPDL$. Then,*

- *not $x_1 R_{\mathcal{M}}(\alpha) x_2$,*
- *$x_1 R_{\mathcal{M}}(\alpha) x_1$ iff $x' R_{\mathcal{M}'}(\alpha) x'$,*
- *$x_1 \in V_{\mathcal{M}}(\psi)$ iff $x' \in V_{\mathcal{M}'}(\psi)$.*

Proof. By induction on α and ψ. Left to the reader. □

Since $x_1 \notin V_{\mathcal{M}}(\phi)$, then $x' \notin V_{\mathcal{M}'}(\phi)$: a contradiction. □

4 Expressivity

In the class of all $*$-separated frames, remark that the formula construct of separating conjunction $(\cdot \circ \cdot)$ and the formula construct of adjoint implication $(\cdot\!\multimap\!\cdot)$ evoked in the introduction can be defined in the language of $PRSPDL$ as follows:

- $(\phi \circ \psi) ::= \langle r_1 \rangle \phi \wedge \langle r_2 \rangle \psi$,
- $(\phi \multimap \psi) ::= [s_2](\langle r_1 \rangle \phi \to \psi)$.

Here are results proving that the program construct $(\cdot \parallel \cdot)$ of parallel composition, the storing programs s_1 and s_2 and the recovering programs r_1 and r_2 cannot be eliminated from the language of $PRSPDL$.

Proposition 4.1 *For all $\|$-free formulas ϕ from the language of $PRSPDL$, $\langle a \parallel a \rangle \top \leftrightarrow \phi$ is not valid in the class of all $*$-separated $*$-deterministic frames for every program variable a.*

Proof. Suppose there exists a $\|$-free formula ϕ from the language of $PRSPDL$ such that $\langle a \parallel a \rangle \top \leftrightarrow \phi$ is valid in the class of all $*$-separated $*$-deterministic frames for some program variable a. Without loss of generality, assume a is the only program variable in ϕ and ϕ contains no propositional variable. Let $\mathcal{F} = (W, R, *)$ and $\mathcal{F}' = (W', R', *')$ be the $*$-separated $*$-deterministic frames defined as follows:

- $W = \{x, y, z, t, u\}$,
- $R(a) = \{(y, z), (y, t)\}$ and otherwise R is the empty function,
- $y * y = \{x\}$, $z * t = \{u\}$ and otherwise $*$ is the empty function,
- $W' = \{x', y', z'_1, z'_2, t'_1, t'_2, u'_1, u'_2\}$,

- $R'(a) = \{(y', z_1'), (y', t_2')\}$ and otherwise R' is the empty function,
- $y' *' y' = \{x'\}$, $z_1' *' t_1' = \{u_1'\}$, $z_2' *' t_2' = \{u_2'\}$ and otherwise $*'$ is the empty function.

Since $\langle a \parallel a\rangle\top \leftrightarrow \phi$ is valid in the class of all $*$-separated $*$-deterministic frames, then $\mathcal{F} \models \langle a \parallel a\rangle\top \leftrightarrow \phi$ and $\mathcal{F}' \models \langle a \parallel a\rangle\top \leftrightarrow \phi$. Let $Z = \{(x,x'),(y,y'),(z,z_1'),(z,z_2'),(t,t_1'),(t,t_2'),(u,u_1'),(u,u_2')\}$. Let $\mathcal{M} = (W,R,*,V)$ be a model on $\mathcal{F}$ and $\mathcal{M}' = (W',R',*',V')$ be the model on $\mathcal{F}'$ corresponding to it with respect to Z. Obviously, $x \in V_{\mathcal{M}}(\langle a \parallel a\rangle\top)$ and $x' \notin V_{\mathcal{M}'}(\langle a \parallel a\rangle\top)$. Since $\mathcal{F} \models \langle a \parallel a\rangle\top \leftrightarrow \phi$ and $\mathcal{F}' \models \langle a \parallel a\rangle\top \leftrightarrow \phi$, then $x \in V_{\mathcal{M}}(\langle a \parallel a\rangle\top \leftrightarrow \phi)$ and $x' \in V_{\mathcal{M}'}(\langle a \parallel a\rangle\top \leftrightarrow \phi)$. Since $x \in V_{\mathcal{M}}(\langle a \parallel a\rangle\top)$ and $x' \notin V_{\mathcal{M}'}(\langle a \parallel a\rangle\top)$, then $x \in V_{\mathcal{M}}(\phi)$ and $x' \notin V_{\mathcal{M}'}(\phi)$.

Claim 4.2 *Let α be a $\parallel$-free program and ψ be a $\parallel$-free formula from the language of $PRSPDL$. For all $v \in W$ and for all $v' \in W'$, if vZv', then*

- *for all $w \in W$, if $vR_{\mathcal{M}}(\alpha)w$, then there exists $w' \in W'$ such that wZw' and $v'R_{\mathcal{M}'}(\alpha)w'$,*
- *for all $w' \in W'$, if $v'R_{\mathcal{M}'}(\alpha)w'$, then there exists $w \in W$ such that wZw' and $vR_{\mathcal{M}}(\alpha)w$,*
- *$v \in V_{\mathcal{M}}(\psi)$ iff $v' \in V_{\mathcal{M}'}(\psi)$.*

Proof. By induction on α and ψ. Left to the reader. □

Since xZx' and $x \in V_{\mathcal{M}}(\phi)$, then $x' \in V_{\mathcal{M}'}(\phi)$: a contradiction. □

Proposition 4.3 *For all storing-free formulas ϕ from the language of $PRSPDL$, $\langle s_i\rangle\top \leftrightarrow \phi$ is not valid in the class of all functional $*$-separated $*$-deterministic frames for every $i \in \{1,2\}$.*

Proof. Suppose there exists a storing-free formula ϕ from the language of $PRSPDL$ such that $\langle s_i\rangle\top \leftrightarrow \phi$ is valid in the class of all functional $*$-separated $*$-deterministic frames for some $i \in \{1,2\}$. Without loss of generality, assume ϕ contains neither program variable, nor propositional variable. Moreover, we can assume $i = 1$. Let $\mathcal{F} = (W,R,*)$ and $\mathcal{F}' = (W',R',*')$ be the functional $*$-separated $*$-deterministic frames defined as follows:

- $W = \{x,y\}$,
- R is the empty function,
- $x * x = \{y\}$ and otherwise $*$ is the empty function,
- $W' = \{x',y'\}$,
- R' is the empty function,
- $*'$ is the empty function.

Since $\langle s_1\rangle\top \leftrightarrow \phi$ is valid in the class of all functional $*$-separated $*$-deterministic frames, then $\mathcal{F} \models \langle s_1\rangle\top \leftrightarrow \phi$ and $\mathcal{F}' \models \langle s_1\rangle\top \leftrightarrow \phi$. Let $\mathcal{M} = (W,R,*,V)$ be a model on $\mathcal{F}$. Let $\mathcal{M}' = (W',R',*',V')$ be the model on $\mathcal{F}'$ defined as follows:

- $V'(p) =$ if $x \in V(p)$, then $\{x'\}$, else $\emptyset$ for every propositional variable p.

Obviously, $x \in V_{\mathcal{M}}(\langle s_1 \rangle \top)$ and $x' \notin V_{\mathcal{M}'}(\langle s_1 \rangle \top)$. Since $\mathcal{F} \models \langle s_1 \rangle \top \leftrightarrow \phi$ and $\mathcal{F}' \models \langle s_1 \rangle \top \leftrightarrow \phi$, then $x \in V_{\mathcal{M}}(\langle s_1 \rangle \top \leftrightarrow \phi)$ and $x' \in V_{\mathcal{M}'}(\langle s_1 \rangle \top \leftrightarrow \phi)$. Since $x \in V_{\mathcal{M}}(\langle s_1 \rangle \top)$ and $x' \notin V_{\mathcal{M}'}(\langle s_1 \rangle \top)$, then $x \in V_{\mathcal{M}}(\phi)$ and $x' \notin V_{\mathcal{M}'}(\phi)$.

Claim 4.4 *Let α be a storing-free program and ψ be a storing-free formula from the language of $PRSPDL$. Then,*

- *not $xR_{\mathcal{M}}(\alpha)y$,*
- *$xR_{\mathcal{M}}(\alpha)x$ iff $x'R_{\mathcal{M}'}(\alpha)x'$,*
- *$x \in V_{\mathcal{M}}(\psi)$ iff $x' \in V_{\mathcal{M}'}(\psi)$.*

Proof. By induction on α and ψ. Left to the reader. □

Since $x \in V_{\mathcal{M}}(\phi)$, then $x' \in V_{\mathcal{M}'}(\phi)$: a contradiction. □

Proposition 4.5 *For all recovering-free formulas ϕ from the language of $PRSPDL$, $[r_i^\star]\langle \top? \parallel \top? \rangle \top \leftrightarrow \phi$ is not valid in the class of all functional $*$-separated $*$-deterministic frames for every $i \in \{1, 2\}$.*

Proof. Suppose there exists a recovering-free formula ϕ from the language of $PRSPDL$ such that $[r_i^\star]\langle \top? \parallel \top? \rangle \top \leftrightarrow \phi$ is valid in the class of all functional $*$-separated $*$-deterministic frames for some $i \in \{1, 2\}$. Let $n = lev(\phi)$. Without loss of generality, assume ϕ contains neither program variable, nor propositional variable. Moreover, we can assume $i = 1$. Let $\mathcal{F} = (W, R, *)$ and $\mathcal{F}' = (W', R', *')$ be the functional $*$-separated $*$-deterministic frames defined as follows:

- $W = \{x, y\} \times \mathbb{N}$,
- R is the empty function,
- $(x, k+1) * (y, k+1) = \{(x, k)\}$ and otherwise $*$ is the empty function,
- $W' = \{x', y'\} \times \{0, \ldots, n\}$,
- R' is the empty function,
- $(x', 1) *' (y', 1) = \{(x', 0)\}$, $\ldots$, $(x', n) *' (y', n) = \{(x', n-1)\}$ and otherwise $*'$ is the empty function.

Since $[r_1^\star]\langle \top? \parallel \top? \rangle \top \leftrightarrow \phi$ is valid in the class of all functional $*$-separated $*$-deterministic frames, then $\mathcal{F} \models [r_1^\star]\langle \top? \parallel \top? \rangle \top \leftrightarrow \phi$ and $\mathcal{F}' \models [r_1^\star]\langle \top? \parallel \top? \rangle \top \leftrightarrow \phi$. Let $Z = \{((x, 0), (x', 0)), \ldots, ((x, n), (x', n)), ((y, 0), (y', 0)), \ldots, ((y, n), (y', n))\}$. Let $\mathcal{M} = (W, R, *, V)$ be a model on $\mathcal{F}$ and $\mathcal{M}' = (W', R', *', V')$ be the model on $\mathcal{F}'$ corresponding to it with respect to Z. Obviously, $(x, 0) \in V_{\mathcal{M}}([r_1^\star]\langle \top? \parallel \top? \rangle \top)$ and $(x', 0) \notin V_{\mathcal{M}'}([r_1^\star]\langle \top? \parallel \top? \rangle \top)$. Since $\mathcal{F} \models [r_1^\star]\langle \top? \parallel \top? \rangle \top \leftrightarrow \phi$ and $\mathcal{F}' \models [r_1^\star]\langle \top? \parallel \top? \rangle \top \leftrightarrow \phi$, then $(x, 0) \in V_{\mathcal{M}}([r_1^\star]\langle \top? \parallel \top? \rangle \top \leftrightarrow \phi)$ and $(x', 0) \in V_{\mathcal{M}'}([r_1^\star]\langle \top? \parallel \top? \rangle \top \leftrightarrow \phi)$. Since $(x, 0) \in V_{\mathcal{M}}([r_1^\star]\langle \top? \parallel \top? \rangle \top)$ and $(x', 0) \notin V_{\mathcal{M}'}([r_1^\star]\langle \top? \parallel \top? \rangle \top)$, then $(x, 0) \in V_{\mathcal{M}}(\phi)$ and $(x', 0) \notin V_{\mathcal{M}'}(\phi)$.

Claim 4.6 *Let α be a recovering-free program from the language of $PRSPDL$. For all $k \in \{0, \ldots, n\}$,*

- $R_{\mathcal{M}}(\alpha)((x,k)) \subseteq \{(x,0),\ldots,(x,k)\}$ *and* $R_{\mathcal{M}'}(\alpha)((x',k)) \subseteq \{(x',0),\ldots,(x',k)\}$,
- $R_{\mathcal{M}}(\alpha)((y,k)) \subseteq \{(x,0),\ldots,(x,k-1)\} \cup \{(y,k)\}$ *and* $R_{\mathcal{M}'}(\alpha)((y',k)) \subseteq \{(x',0),\ldots,(x',k-1)\} \cup \{(y',k)\}$.

Proof. By induction on α. Left to the reader. $\square$

Claim 4.7 *Let α be a recovering-free program and ψ be a recovering-free formula from the language of $PRSPDL$. For all $k \in \{0,\ldots,n\}$, if $k+lev(\alpha) \leq n$ and $k+lev(\psi) \leq n$, then*

- $(x,k)R_{\mathcal{M}}(\alpha)(x,l)$ *iff* $(x',k)R_{\mathcal{M}'}(\alpha)(x',l)$,
- $(y,k)R_{\mathcal{M}}(\alpha)(x,l)$ *iff* $(y',k)R_{\mathcal{M}'}(\alpha)(x',l)$,
- $(y,k)R_{\mathcal{M}}(\alpha)(y,l)$ *iff* $(y',k)R_{\mathcal{M}'}(\alpha)(y',l)$,
- $(x,k) \in V_{\mathcal{M}}(\psi)$ *iff* $(x',k) \in V_{\mathcal{M}'}(\psi)$,
- $(y,k) \in V_{\mathcal{M}}(\psi)$ *iff* $(y',k) \in V_{\mathcal{M}'}(\psi)$.

Proof. By induction on α and ψ. Left to the reader. $\square$

Since $(x,0) \in V_{\mathcal{M}}(\phi)$, then $(x',0) \in V_{\mathcal{M}'}(\phi)$: a contradiction. $\square$

Now, let us extend $PRSPDL$ with the program construct $(\cdot \cap \cdot)$ of intersection. In this variant, intersection $(\alpha \cap \beta)$ of programs α and β corresponds to the intersection of the accessibility relations $R(\alpha)$ and $R(\beta)$. We have the following:

Proposition 4.8 *Let α, β be programs from the language of $PRSPDL$ extended with $(\cdot\cap\cdot)$. For all models $\mathcal{M}$, $R_{\mathcal{M}}(\alpha \parallel \beta) = R_{\mathcal{M}}((r_1;\alpha;s_1)\cap(r_2;\beta;s_2))$.*

Proof. Left to the reader. $\square$

Hence, the program construct $(\cdot \parallel \cdot)$ of parallel composition can be eliminated from the language of $PRSPDL$ extended with $(\cdot \cap \cdot)$. Nevertheless,

Proposition 4.9 *For all formulas ϕ from the language of $PRSPDL$, $\langle a \cap b\rangle\top \leftrightarrow \phi$ is not valid in the class of all functional $*$-separated $*$-deterministic frames for every distinct program variables a, b.*

Proof. Suppose there exists a formula ϕ from the language of $PRSPDL$ such that $\langle a \cap b\rangle\top \leftrightarrow \phi$ is valid in the class of all functional $*$-separated $*$-deterministic frames for some distinct program variables a, b. Without loss of generality, assume a, b are the only program variables in ϕ and ϕ contains no propositional variable. Let $\mathcal{F} = (W, R, *)$ and $\mathcal{F}' = (W', R', *')$ be the functional $*$-separated $*$-deterministic frames s defined as follows:

- $W = \{x, y\}$,
- $R(a) = \{(x,y)\}$, $R(b) = \{(x,y)\}$ and otherwise R is the empty function,
- $*$ is the empty function,
- $W' = \{x', y'_1, y'_2\}$,

- $R'(a) = \{(x', y'_1)\}$, $R'(b) = \{(x', y'_2)\}$ and otherwise R' is the empty function,
- $*'$ is the empty function.

Since $\langle a \cap b \rangle \top \leftrightarrow \phi$ is valid in the class of all functional $*$-separated $*$-deterministic frames, then $\mathcal{F} \models \langle a \cap b \rangle \top \leftrightarrow \phi$ and $\mathcal{F}' \models \langle a \cap b \rangle \top \leftrightarrow \phi$. Let $Z = \{(x, x'), (y, y'_1), (y, y'_2)\}$. Let $\mathcal{M} = (W, R, *, V)$ be a model on $\mathcal{F}$ and $\mathcal{M}' = (W', R', *', V')$ be the model on $\mathcal{F}'$ corresponding to it with respect to Z. Obviously, $x \in V_{\mathcal{M}}(\langle a \cap b \rangle \top)$ and $x' \notin V_{\mathcal{M}'}(\langle a \cap b \rangle \top)$. Since $\mathcal{F} \models \langle a \cap b \rangle \top \leftrightarrow \phi$ and $\mathcal{F}' \models \langle a \cap b \rangle \top \leftrightarrow \phi$, then $x \in V_{\mathcal{M}}(\langle a \cap b \rangle \top \leftrightarrow \phi)$ and $x' \in V_{\mathcal{M}'}(\langle a \cap b \rangle \top \leftrightarrow \phi)$. Since $x \in V_{\mathcal{M}}(\langle a \cap b \rangle \top)$ and $x' \notin V_{\mathcal{M}'}(\langle a \cap b \rangle \top)$, then $x \in V_{\mathcal{M}}(\phi)$ and $x' \notin V_{\mathcal{M}'}(\phi)$.

Claim 4.10 *Let α be a program and ψ be a formula from the language of $PRSPDL$. For all $v \in W$ and for all $v' \in W'$, if vZv', then*

- *for all $w \in W$, if $vR_{\mathcal{M}}(\alpha)w$, then there exists $w' \in W'$ such that wZw' and $v'R_{\mathcal{M}'}(\alpha)w'$,*
- *for all $w' \in W'$, if $v'R_{\mathcal{M}'}(\alpha)w'$, then there exists $w \in W$ such that wZw' and $vR_{\mathcal{M}}(\alpha)w$,*
- *$v \in V_{\mathcal{M}}(\psi)$ iff $v' \in V_{\mathcal{M}'}(\psi)$.*

Proof. By induction on α and ψ. Left to the reader. □

Since xZx' and $x \in V_{\mathcal{M}}(\phi)$, then $x' \in V_{\mathcal{M}'}(\phi)$: a contradiction. □

Hence, the program construct $(\cdot \cap \cdot)$ of intersection cannot be defined in the language of $PRSPDL$.

5 Decidability

Let $\mathcal{L}^{s_1,s_2}_{PDL}$ be the set of all $\|$-free recovering-free formulas. Let $\mathcal{C}_{*sep}$ be the class of all $*$-separated frames and $\mathcal{C}^{*det}_{*sep}$ be the class of all $*$-separated $*$-deterministic frames. The tree model property of PDL enables us to prove the following:

Proposition 5.1 (i) *$VAL(\mathcal{L}^{s_1,s_2}_{PDL}, \mathcal{C}_{*sep})$ is $EXPTIME$-complete.*
(ii) *$VAL(\mathcal{L}^{s_1,s_2}_{PDL}, \mathcal{C}^{*det}_{*sep})$ is $EXPTIME$-complete.*

Proof. The key thing to note about $\|$-free recovering-free formulas is the following:

Claim 5.2 *Let $\phi \in \mathcal{L}^{s_1,s_2}_{PDL}$. The following conditions are equivalent:*

a) ϕ, where s_1 and s_2 are considered as ordinary program variables, is satisfied in a PDL-frame.

b) ϕ, where s_1 and s_2 are considered as ordinary program variables, is satisfied in a tree-like PDL-frame.

c) ϕ, where s_1 and s_2 are considered as storing programs, is satisfied in a $*$-separated $*$-deterministic $PRSPDL$-frame.

d) ϕ, where s_1 and s_2 are considered as storing programs, is satisfied in a $*$-separated $PRSPDL$-frame.

Proof. *a*) $\Rightarrow$ *b*) Suppose ϕ, where s_1 and s_2 are considered as ordinary program variables, is satisfied in a PDL-frame $\mathcal{F} = (W, R)$. Hence, there exists a model $\mathcal{M} = (W, R, V)$ on $\mathcal{F}$ and there exists $x \in W$ such that $x \in V_{\mathcal{M}}(\phi)$. Let $\mathcal{F}' = (W', R')$ be the *Unravelling* of $\mathcal{F}$ around x and $\mathcal{M}' = (W', R', V')$ be the model on $\mathcal{F}'$ corresponding to $\mathcal{M}$. See [6, Pages 63, 218 and 219] for precise definitions. Obviously, $\mathcal{F}'$ is a tree-like PDL-frame. Since $x \in V_{\mathcal{M}}(\phi)$, by [6, Proposition 2.14 and Lemma 4.52], then $(x) \in V_{\mathcal{M}'}(\phi)$. Thus, ϕ, where s_1 and s_2 are considered as ordinary program variables, is satisfied in a tree-like PDL-frame.

b) $\Rightarrow$ *c*) Suppose ϕ, where s_1 and s_2 are considered as ordinary program variables, is satisfied in a tree-like PDL-frame $\mathcal{F} = (W, R)$. Hence, there exists a model $\mathcal{M} = (W, R, V)$ on $\mathcal{F}$ and there exists $x \in W$ such that $x \in V_{\mathcal{M}}(\phi)$. Let $\mathcal{F}' = (W', R', *')$ be the $*$-separated $*$-deterministic $PRSPDL$-frame defined as follows:

- $W' = W \cup \{(y, z, i)\colon\ y, z \in W,\ i \in \{1, 2\} \text{ and } yR(s_i)z\}$,
- $R'(a) = R(a)$ for every program variable a,
- $y *' (y, z, 1) = \{z\}$ for every $(y, z, 1) \in W'$, $(y, z, 2) *' y = \{z\}$ for every $(y, z, 2) \in W'$ and otherwise $*'$ is the empty function.

Let $\mathcal{M}' = (W', R', *', V')$ be the model on $\mathcal{F}'$ defined as follows:

- $V'(p) = V(p)$ for every propositional variable p.

Claim 5.3 *Let* $\psi \in \mathcal{L}_{PDL}^{s_1,s_2}$. *For all* $y \in W$, $y \in V_{\mathcal{M}}(\psi)$ *iff* $y \in V_{\mathcal{M}'}(\psi)$.

Proof. By induction on ψ. Left to the reader. □

Since $x \in V_{\mathcal{M}}(\phi)$, then $x \in V_{\mathcal{M}'}(\phi)$. Thus, ϕ, where s_1 and s_2 are considered as storing programs, is satisfied in a $*$-separated $*$-deterministic $PRSPDL$-frame.

c) $\Rightarrow$ *d*) Obvious.

d) $\Rightarrow$ *a*) Suppose ϕ, where s_1 and s_2 are considered as storing programs, is satisfied in a $*$-separated $PRSPDL$-frame $\mathcal{F} = (W, R, *)$. Hence, there exists a model $\mathcal{M} = (W, R, *, V)$ on $\mathcal{F}$ and there exists $x \in W$ such that $x \in V_{\mathcal{M}}(\phi)$. Let $\mathcal{F}' = (W', R')$ be the PDL-frame defined as follows:

- $W' = W$,
- $R'(a) = R(a)$ for every program variable a,
- $R'(s_1) = \{(x, y)\colon\ x, y, z \in W \text{ and } y \in x * z\}$,
- $R'(s_2) = \{(x, y)\colon\ x, y, z \in W \text{ and } y \in z * x\}$.

Let $\mathcal{M}' = (W', R', V')$ be the model on $\mathcal{F}'$ defined as follows:

- $V'(p) = V(p)$ for every propositional variable p.

Claim 5.4 *Let* $\psi \in \mathcal{L}_{PDL}^{s_1,s_2}$. *For all* $y \in W$, $y \in V_{\mathcal{M}}(\psi)$ *iff* $y \in V_{\mathcal{M}'}(\psi)$.

Proof. By induction on ψ. Left to the reader. □

Since $x \in V_{\mathcal{M}}(\phi)$, then $x \in V_{\mathcal{M}'}(\phi)$. Thus, ϕ, where s_1 and s_2 are considered as ordinary program variables, is satisfied in a PDL-frame. □

Since satisfiability in a PDL-frame of $\mathcal{L}_{PDL}^{s_1,s_2}$-formulas where s_1 and s_2 are considered as ordinary program variables is $EXPTIME$-complete [11,24], then satisfiability in a $*$-separated $PRSPDL$-frame and satisfiability in a $*$-separated $*$-deterministic $PRSPDL$-frame of $\mathcal{L}_{PDL}^{s_1,s_2}$-formulas where s_1 and s_2 are considered as storing programs are $EXPTIME$-complete. Hence, $VAL(\mathcal{L}_{PDL}^{s_1,s_2}, \mathcal{C}_{*sep})$ and $VAL(\mathcal{L}_{PDL}^{s_1,s_2}, \mathcal{C}_{*sep}^{*det})$ are $EXPTIME$-complete. $\square$

Let $\mathcal{L}_{;}^{s_1,s_2}$ be the set of all ?-free $\star$-free $\|$-free recovering-free formulas. Claim 5.2 enables us to prove the following:

Proposition 5.5 1) *$VAL(\mathcal{L}_{;}^{s_1,s_2}, \mathcal{C}_{*sep})$ is $PSPACE$-complete.*

2) *$VAL(\mathcal{L}_{;}^{s_1,s_2}, \mathcal{C}_{*sep}^{*det})$ is $PSPACE$-complete.*

Proof. By Claim 5.2, ϕ, where s_1 and s_2 are considered as ordinary program variables, is satisfied in a PDL-frame iff ϕ, where s_1 and s_2 are considered as storing programs, is satisfied in a $*$-separated $*$-deterministic $PRSPDL$-frame iff ϕ, where s_1 and s_2 are considered as storing programs, is satisfied in a $*$-separated $PRSPDL$-frame for every $\phi \in \mathcal{L}_{;}^{s_1,s_2}$. Since satisfiability in a PDL-frame of $\mathcal{L}_{;}^{s_1,s_2}$-formulas where s_1 and s_2 are considered as ordinary program variables is $PSPACE$-complete [20], then satisfiability in a $*$-separated $PRSPDL$-frame and satisfiability in a $*$-separated $*$-deterministic $PRSPDL$-frame of $\mathcal{L}_{;}^{s_1,s_2}$-formulas where s_1 and s_2 are considered as storing programs are $PSPACE$-complete. Hence, $VAL(\mathcal{L}_{;}^{s_1,s_2}, \mathcal{C}_{*sep})$ and $VAL(\mathcal{L}_{;}^{s_1,s_2}, \mathcal{C}_{*sep}^{*det})$ are $PSPACE$-complete. $\square$

6 Undecidability

Together with the decidability of PDL with intersection obtained by Danecki [9], Propositions 4.8 and 4.9 seems to indicate that $PRSPDL$ is decidable. It is interesting to observe that this assertion is false. Let $\mathcal{L}_{PDL}^{\|,r_1,r_2}$ be the set of all storing-free formulas. Solving an open problem put forward in [5], with the aid of the $\mathbb{N} \times \mathbb{N}$ recurring tiling problem, let us prove the following:

Proposition 6.1 *$VAL(\mathcal{L}_{PDL}^{\|,r_1,r_2}, \mathcal{C})$ is Π_1^1-hard for the following classes $\mathcal{C}$ of frames:*

- *the class $\mathcal{C}_{fun,*sep}^{*det,*ser}$ of all functional $*$-separated $*$-deterministic $*$-serial frames,*
- *the class $\mathcal{C}_{fun,*sep}^{*det}$ of all functional $*$-separated $*$-deterministic frames,*
- *the class $\mathcal{C}_{fun,*sep}^{*ser}$ of all functional $*$-separated $*$-serial frames,*
- *the class $\mathcal{C}_{*sep}^{*det,*ser}$ of all $*$-separated $*$-deterministic $*$-serial frames,*
- *the class $\mathcal{C}_{fun,*sep}$ of all functional $*$-separated frames,*
- *the class $\mathcal{C}_{*sep}^{*det}$ of all $*$-separated $*$-deterministic frames,*
- *the class $\mathcal{C}_{*sep}^{*ser}$ of all $*$-separated $*$-serial frames,*
- *the class $\mathcal{C}_{*sep}$ of all $*$-separated frames.*

Proof. Let $\mathcal{C}$ be one of the classes of frames considered in Proposition 6.1. A tile type t is a square, fixed in orientation, each side of which has a color: $left(t)$, $right(t)$, $down(t)$ and $up(t)$. A finite set T of tile types is said to tile $\mathbb{N} \times \mathbb{N}$ iff there exists a function f from $\mathbb{N} \times \mathbb{N}$ into T such that for all $x, y \in \mathbb{N}$, $right(f(x,y)) = left(f(x+1,y))$ and $up(f(x,y)) = down(f(x,y+1))$. The $\mathbb{N} \times \mathbb{N}$ recurring tiling problem is the following decision problem:

- input: a finite set T of tile types which includes some distinguished tile type t_1;
- output: determine whether T can tile $\mathbb{N} \times \mathbb{N}$ in such a way that t_1 occurs infinitely often in the first row.

It is a well-known fact that the $\mathbb{N} \times \mathbb{N}$ recurring tiling problem is Σ_1^1-hard [18]. Given pairwise distinct tile types $t_1, \ldots, t_n$, let $p_1, \ldots, p_n$ be pairwise distinct propositional variables. We associate to $t_1, \ldots, t_n$, the conjunction $\phi(t_1, \ldots, t_n)$ of the following formulas:

$B1$ $[r_1^\star \parallel r_2^\star]\langle r_1 \parallel \top?\rangle\top$;

$B2$ $[r_1^\star \parallel r_2^\star]\langle \top? \parallel r_2\rangle\top$;

$B3$ $[r_1^\star \parallel r_2^\star]\neg(p_i \wedge p_j)$ for every $i, j \in \{1, \ldots, n\}$ such that $i \neq j$;

$B4$ $[r_1^\star \parallel r_2^\star](p_1 \vee \ldots \vee p_n)$;

$B5$ $[r_1^\star \parallel r_2^\star](p_i \rightarrow [\top? \parallel \top?]p_i)$ for every $i \in \{1, \ldots, n\}$;

$B6$ $[r_1^\star \parallel r_2^\star](p_i \rightarrow \langle r_1 \parallel \top?\rangle(p_{k_1} \vee \ldots \vee p_{k_l}))$ for every $i \in \{1, \ldots, n\}$, $t_{k_1}, \ldots, t_{k_l}$ being the tile types in $t_1, \ldots, t_n$ horizontally matching with t_i;

$B7$ $[r_1^\star \parallel r_2^\star](p_i \rightarrow \langle \top? \parallel r_2\rangle(p_{k_1} \vee \ldots \vee p_{k_l}))$ for every $i \in \{1, \ldots, n\}$, $t_{k_1}, \ldots, t_{k_l}$ being the tile types in $t_1, \ldots, t_n$ vertically matching with t_i.

Let $\psi(t_1, \ldots, t_n) ::= \langle \top? \parallel \top?\rangle\top \wedge \phi(t_1, \ldots, t_n) \wedge [r_1^\star \parallel \top?]\langle r_1^\star \parallel \top?\rangle p_1$.

Claim 6.2 *The following conditions are equivalent:*

a) $\{t_1, \ldots, t_n\}$ *can tile* $\mathbb{N} \times \mathbb{N}$ *in such a way that* t_1 *occurs infinitely often in the first row.*

b) $\psi(t_1, \ldots, t_n)$ *is satisfied in* $\mathcal{C}$.

Proof. *a)* $\Rightarrow$ *b)* Suppose $\{t_1, \ldots, t_n\}$ can tile $\mathbb{N} \times \mathbb{N}$ in such a way that t_1 occurs infinitely often in the first row. Hence, there exists a function f from $\mathbb{N} \times \mathbb{N}$ into T such that for all $x, y \in \mathbb{N}$, $right(f(x,y)) = left(f(x+1,y))$ and $up(f(x,y)) = down(f(x,y+1))$. Moreover, for all $x \in \mathbb{N}$, there exists $z \in \mathbb{N}$ such that $x \leq z$ and $f(z, 0) = t_1$. Let $\mathcal{F} = (W, R, *)$ be the functional $*$-separated $*$-deterministic $*$-serial $PRSPDL$-frame defined as follows:

- $W = (\mathbb{N} \times \mathbb{N}) \cup (\mathbb{N} \times \{l_1, l_2, l_3, l_4\})$ where l_1, l_2, l_3, l_4 are new distinct elements,
- R is the empty function,
- $*$ is a one-to-one correspondence between the elements of $W \times W$ and the singletons over W such that $(x, l_1) * (y, l_2) = \{(x, y)\}$, $(x+1, l_1) * (x, l_3) = \{(x, l_1)\}$ and $(y, l_4) * (y+1, l_2) = \{(y, l_2)\}$.

Let $\mathcal{M} = (W, R, *, V)$ be the model on $\mathcal{F}$ defined as follows:

- $V(p_i) = \{(x, y)\colon f(x, y) = t_i\}$ and otherwise V is the empty function.

Obviously, for all $x, y \in \mathbb{N}$, $(x + 1, y)$ is the only state in W accessible from (x, y) by means of $R_{\mathcal{M}}(r_1 \parallel \top?)$ and $(x, y + 1)$ is the only state in W accessible from (x, y) by means of $R_{\mathcal{M}}(\top? \parallel r_2)$. Thus, $(0, 0) \in V_{\mathcal{M}}(\psi(t_1, \dots, t_n))$. Therefore, $\psi(t_1, \dots, t_n)$ is satisfied in a functional $*$-separated $*$-deterministic $*$-serial $PRSPDL$-frame. Consequently, $\psi(t_1, \dots, t_n)$ is satisfied in $\mathcal{C}$.

b) $\Rightarrow$ *a)* Suppose $\psi(t_1, \dots, t_n)$ is satisfied in $\mathcal{C}$. Hence, $\psi(t_1, \dots, t_n)$ is satisfied in a $*$-separated $PRSPDL$-frame $\mathcal{F} = (W, R, *)$. Thus, there exists a model $\mathcal{M} = (W, R, *, V)$ on $\mathcal{F}$ and there exists $u \in W$ such that $u \in V_{\mathcal{M}}(\psi(t_1, \dots, t_n))$. Obviously, thanks to the formulas $B1$ and $B2$, for all $x, y \in \mathbb{N}$, there exists $v \in W$ such that $uR_{\mathcal{M}}(r_1^x \parallel r_2^y)v$; the set of all such v will be denoted $P(x, y)$. Moreover, thanks to the formulas $B3$, $B4$ and $B5$, for all $x, y \in \mathbb{N}$, there exists $i \in \{1, \dots, n\}$ such that $P(x, y) \subseteq V(p_i)$ and for all $j \in \{1, \dots, n\}$, if $i \neq j$, then $P(x, y) \cap V(p_j) = \emptyset$. In other respect, thanks to the formulas $B6$ and $B7$, for all $x, y \in \mathbb{N}$ and for all $i, j \in \{1, \dots, n\}$, if $P(x, y) \subseteq V(p_i)$ and $P(x + 1, y) \subseteq V(p_j)$, then $right(t_i) = left(t_j)$ and if $P(x, y) \subseteq V(p_i)$ and $P(x, y + 1) \subseteq V(p_j)$, then $up(t_i) = down(t_j)$. Finally, thanks to the formula $[r_1^\star \parallel \top?]\langle r_1^\star \parallel \top?\rangle p_1$, for all $x \in \mathbb{N}$, there exists $z \in \mathbb{N}$ such that $x \leq z$ and $P(z, 0) \subseteq V(p_1)$. Let f be the function from $\mathbb{N} \times \mathbb{N}$ into $\{t_1, \dots, t_n\}$ defined as follows:

- $f(x, y) = t_i$ iff $P(x, y) \subseteq V(p_i)$.

Obviously, for all $x, y \in \mathbb{N}$, $right(f(x, y)) = left(f(x+1, y))$ and $up(f(x, y)) = down(f(x, y + 1))$. Moreover, for all $x \in \mathbb{N}$, there exists $z \in \mathbb{N}$ such that $x \leq z$ and $f(z, 0) = t_1$. Therefore, $\{t_1, \dots, t_n\}$ can tile $\mathbb{N} \times \mathbb{N}$ in such a way that t_1 occurs infinitely often in the first row. $\square$

Hence, the $\mathbb{N} \times \mathbb{N}$ recurring tiling problem is reducible to satisfiability in the class $\mathcal{C}$ of $\mathcal{L}_{PDL}^{\parallel, r_1, r_2}$-formulas. Since the $\mathbb{N} \times \mathbb{N}$ recurring tiling problem is Σ_1^1-hard [18], satisfiability in the class $\mathcal{C}$ of $\mathcal{L}_{PDL}^{\parallel, r_1, r_2}$-formulas is Σ_1^1-hard. Thus, $VAL(\mathcal{L}_{PDL}^{\parallel, r_1, r_2}, \mathcal{C})$ is Π_1^1-hard. $\square$

Corollary 6.3 *$VAL(\mathcal{L}_{PDL}^{\parallel, r_1, r_2}, \mathcal{C})$ is Π_1^1-complete for all classes $\mathcal{C}$ of frames considered in Proposition 6.1.*

Proof. It suffices to prove that if a $PRSPDL$-formula is satisfied in a frame in $\mathcal{C}$, then it is satisfied in a finite or countable frame in $\mathcal{C}$. By means of the so-called *Standard Translation,* one can prove that $PRSPDL$ is a fragment of $L_{\omega_1 \omega}$, the infinitary logic in which one is allowed to consider countable conjunctions in addition to the usual first-order constructs. See [6, Pages 83–86 and 496] for precise definitions. By the Löwenheim-Skolem theorem for $L_{\omega_1 \omega}$, if the standard translation of a $PRSPDL$-formula is satisfied in a frame in $\mathcal{C}$, then it is satisfied in a finite or countable frame in $\mathcal{C}$. Hence, if a $PRSPDL$-formula is satisfied in a frame in $\mathcal{C}$, then it is satisfied in a finite or countable frame in $\mathcal{C}$. $\square$

7 Conclusion

We present our computability results in the following tables.

	$\mathcal{C}^{*det,*ser}_{fun,*sep}$	$\mathcal{C}^{*det}_{fun,*sep}$	$\mathcal{C}^{*ser}_{fun,*sep}$	$\mathcal{C}^{*det,*ser}_{*sep}$
$\mathcal{L}^{s_1,s_2}_{PDL}$				
$\mathcal{L}^{s_1,s_2}_{;}$				
$\mathcal{L}^{\parallel,r_1,r_2}_{PDL}$	Π^1_1-c	Π^1_1-c	Π^1_1-c	Π^1_1-c

	$\mathcal{C}_{fun,*sep}$	$\mathcal{C}^{*det}_{*sep}$	$\mathcal{C}^{*ser}_{*sep}$	$\mathcal{C}_{*sep}$
$\mathcal{L}^{s_1,s_2}_{PDL}$		$EXPTIME$-c		$EXPTIME$-c
$\mathcal{L}^{s_1,s_2}_{;}$		$PSPACE$-c		$PSPACE$-c
$\mathcal{L}^{\parallel,r_1,r_2}_{PDL}$	Π^1_1-c	Π^1_1-c	Π^1_1-c	Π^1_1-c

Let $\mathcal{C}$ be one of the classes of frames considered in the above tables.

As a consequence of Corollary 6.3, $VAL(\mathcal{L}^{\parallel,r_1,r_2}_{PDL}, \mathcal{C})$ is Π^1_1-complete. Nevertheless, one may try to axiomatize $VAL(\mathcal{L}^{\parallel,r_1,r_2}_{PDL}, \mathcal{C})$ by means of an infinitary derivation rule similar to the one used in [4]. The result stated in Proposition 4.8 suggests that an unorthodox derivation rule similar to the one used in [3] could be considered as well.

As for the set $\mathcal{L}^{\parallel,s_1,s_2}_{PDL}$ of all recovering-free formulas, the decidability/undecidability status of $VAL(\mathcal{L}^{\parallel,s_1,s_2}_{PDL}, \mathcal{C})$ is not known.

In other respect, seeing that in a frame $\mathcal{F} = (W, R, *)$, W is to be regarded as the set of all possible states in a computation process, it seems natural to consider the restriction $\mathcal{C}\,|\mathrm{wf}$ of $\mathcal{C}$ to those frames $\mathcal{F} = (W, R, *)$ in which the transitive closure of the binary relation $\longrightarrow_{\mathcal{F}}$ defined as follows is well-founded:

- $x \longrightarrow_{\mathcal{F}} y$ iff there exists $z \in W$ such that either $x \in y * z$, or $x \in z * y$.

Remark that the transitive closures of the binary relations $\longrightarrow_{\mathcal{F}_1}$, $\longrightarrow_{\mathcal{F}_2}$ and $\longrightarrow_{\mathcal{F}_3}$ associated to the frames $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$ considered in Example 2.2 are well-founded. The computability status of $VAL(\mathcal{L}^{\parallel,r_1,r_2}_{PDL}, \mathcal{C}\,|\mathrm{wf})$ is not known.

Finally, following the line of reasoning suggested in [12], the accessibility relation associated to $(\alpha \parallel \beta)$ can be defined as follows in the class of all $*$-deterministic frames:

- whenever x and y are related via $R(\alpha)$ and x and z are related via $R(\beta)$, x and $y * z$ are related via $R(\alpha \parallel \beta)$.

Seeing that this variant of $PRSPDL$ is appealing in computer science, especially in system specification and program construction [13], it seems natural to consider, with respect to it, the computability status of satisfiability in the class $\mathcal{C}$ of $\mathcal{L}^{\parallel,r_1,r_2}_{PDL}$-formulas. The equivalence, in the class of all $*$-separated frames, between $(\alpha \parallel \beta)$ — when interpreted by Benevides *et al.* [5] — and $((r_1;\alpha) \parallel (r_2;\beta))$ — when interpreted by Frias [12] and Frias *et al.* [13] — suggests that satisfiability in the class $\mathcal{C}$ of $\mathcal{L}^{\parallel,r_1,r_2}_{PDL}$-formulas is highly undecidable.

Acknowledgements

Special acknowledgement is heartly granted to Joseph Boudou who made several helpful comments for improving the correctness and the readability of this article. The work of Philippe Balbiani was partially supported by the "French National Research Agency" (contract ANR-11-BS02-011) and the "Bulgarian National Science Fund" (contract DID02/32/2009). The work of Tinko Tinchev was partially supported by the "Centre international de mathématiques et d'informatique" (contract ANR-11-LABX-0040-CIMI within the program ANR-11-IDEX-0002-02).

References

[1] Balbiani, P., Boudou, J.: Decidability of iteration-free PDL with parallel composition. Submitted for publication.

[2] Balbiani, P., Boudou, J.: Iteration-free PDL with storing, recovering and parallel composition: a complete axiomatization. Submitted for publication.

[3] Balbiani, P., Vakarelov, D.: Iteration-free PDL with intersection: a complete axiomatization. Fundamenta Informaticæ**45** (2001) 173–194.

[4] Balbiani, P., Vakarelov, D.: PDL with intersection of programs: a complete axiomatization. Journal of Applied Non-Classical Logics **13** (2003) 231–276.

[5] Benevides, M., de Freitas, R., Viana, P.: Propositional dynamic logic with storing, recovering and parallel composition. Electronic Notes in Theoretical Computer Science **269** (2011) 95–107.

[6] Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press (2001).

[7] Brotherston, J., Calcagno, C.: Classical BI: its semantics and proof theory. Logical Methods in Computer Science **6** (2010) 1–42.

[8] Courtault, J.-R., Galmiche, D.: A modal BI logic for dynamic resource properties. In Artemov, S., Nerode, A. (Editors): Logical Foundations of Computer Science. Springer (2013) 134–148.

[9] Danecki, R.: Nondeterministic propositional dynamic logic with intersection is decidable. In Skowron, A. (Editor): Computation Theory. Springer (1985) 34–53.

[10] Fariñas del Cerro, L., Orłowska, E.: DAL — a logic for data analysis. Theoretical Computer Science **36** (1985) 251–264.

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

[12] Frias, M.: Fork Algebras in Algebra, Logic and Computer Science. World Scientific (2002).

[13] Frias, M., Veloso, P., Baum, G.: Fork algebras: past, present and future. Journal of Relational Methods in Computer Science **1** (2004) 181–216.

[14] Galmiche, D., Larchey-Wendling, D.: Expressivity properties of Boolean BI through relational models. In Arun-Kumar, S., Garg, N. (Editors): FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science. Springer (2006) 357–368.

[15] Gargov, G., Passy, S.: A note on Boolean modal logic. In Petkov, P. (Editor): Mathematical Logic. Plenum Press (1990) 299–309.

[16] Goldblatt, R.: Logics of Time and Computation. Center for the Study of Language and Information (1987).

[17] Göller, S., Lohrey, M., Lutz, C.: PDL with intersection and converse: satisfiability and infinite-state model checking. The Journal of Symbolic Logic **74** (2009) 279–314.

[18] Harel, D.: Recurring dominoes: making the highly undecidable highly understandable. Annals of Discrete Mathematics **24** (1985) 51–72.

[19] Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press (2000).

[20] Ladner, R.: The computational complexity of provability in systems of modal propositional logic. SIAM Journal on Computing **6** (1977) 467–480.

[21] Massacci, F.: Decision procedures for expressive description logics with intersection, composition, converse of roles and role identity. In Nebel, B. (Editor): Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence. Morgan Kaufmann (2001) 193–198.

[22] Mirkowska, G.: PAL — propositional algorithmic logic. Fundamenta Informaticæ**4** (1981) 675–760.

[23] O'Hearn, P., Pym, D.: The logic of bunched implications. The Bulletin of Symbolic Logic **5** (1999) 215–244.

[24] Pratt, V.: Models of program logics. In: 20th Annual Symposium on Foundations of Computer Science. IEEE (1979) 115–122.

[25] Reynolds, VJ.: Separation logic: a logic for shared mutable data structures. In: 17th Annual IEEE Symposium on Logic in Computer Science. IEEE (2002) 55–74.