Exponential-Size Model Property for PDL with Separating Parallel Composition

Joseph Boudou(✉)

IRIT, Toulouse University, Toulouse, France
joseph.boudou@irit.fr

Abstract. Propositional dynamic logic is extended with a parallel program having a separating semantic: the program $(\alpha \parallel \beta)$ executes α and β on two substates of the current state. We prove that when the composition of two substates is deterministic, the logic has the exponential-size model property. The proof is by a piecewise filtration using an adaptation of the Fischer-Ladner closure. We conclude that the satisfiability of the logic is decidable in NEXPTIME.

1 Introduction

Propositional dynamic logic (PDL) is a multi-modal logic designed to reason about behaviors of programs [10]. With each program α we associate a modal operator $[\alpha]$, formulas $[\alpha]\varphi$ being read "all executions of α from the current state lead to a state where φ holds". The set of programs is structured by some operators: sequential composition $(\alpha ; \beta)$ of programs α and β executes β after α; nondeterministic choice $(\alpha \cup \beta)$ of program α and β executes α or β, nondeterministically; test $\varphi?$ on formula φ does nothing but can be executed only if the current state satisfies φ; iteration α^* of program α executes α a nondeterministic number of times. A limitation of PDL is the lack of a construct to reason about concurrency.

Different extensions of PDL have been devised to overcome this limitation, for instance interleaving PDL [1], PDL with intersection [11] and the concurrent dynamic logic [16]. PDL with storing, recovering and parallel composition (PRSPDL) [4] is another extension of PDL for concurrency. The key difference is that in PRSPDL, the program $(\alpha \parallel \beta)$ executes α and β in parallel *on two substates* of the current state. Hence, $(\alpha \parallel \beta)$ being executable at some state does not imply that α or β is executable at that state. Moreover, since states can be separated in substates (and merged back), PRSPDL is related to the Boolean logic of bunched implication (BBI) [17]. Indeed, a multiplicative conjunction semantically similar to the one found in BBI can be defined in PRSPDL. Thus PRSPDL can be compared to the classical version of the multi-modal logic of bunched implication (MBIc) [7], a difference being that MBIc has only the parallel composition as program constructs, limiting its expressive power [2].

This work was supported by the "French National Research Agency" (DynRes contract ANR-11-BS02-011).

DOI: 10.1007/978-3-662-48057-1 10

The combination of separation and concurrency provided by the parallel construct of PRSPDL suggests some interesting applications. For instance, a dynamic and concurrent logic on heaps of memory akin to separation logics [8,18], may be envisioned. Moreover, as PDL has been adapted to different contexts, a separating parallel composition may be of interest in some of them. For instance, in dynamic epistemic logics [9], the parallel epistemic action $(!\varphi \parallel !\psi)$ could mean that φ is announced to a group of agents and ψ is announced to the agents not in that group. Despite this potential, there has been almost no complexity analysis of dynamic logic with separating parallel composition. The complexity of PRSPDL is not studied in [4]. In [3], PRSPDL interpreted over frames where there is at most one decomposition of any state into substates is proved to be highly undecidable.

In this paper, we study the complexity of PDL with separating parallel composition (PPDL). The language of this logic is the fragment of PRSPDL without the store and recover programs which allow to access substates directly. We focus on the class of $\lhd$-deterministic frames where the composition of substates is deterministic: there is at most one way to merge two states. This restriction is quite natural and has been studied in many logics with separation like separation logics, ambient logics [6], BBI [13] and arrow logics [14]. We show that for PPDL, $\lhd$-determinism conveys some interesting properties, notably a strong finite model property leading to a complexity upper bound of NEXPTIME for the satisfiability problem. This result contrasts with the 2EXPTIME complexity of other dynamic logics with concurrency like PDL with intersection [12] or interleaving PDL [15]. To prove this result, we provide nontrivial adaptations of existing methods (Fischer-Ladner closure and model unraveling) along with some new concepts (placeholders, marking functions and the neat model property).

The paper is structured as follows. In the next section, the language and semantic of PPDL is formally defined. In Sect. 3, the problem of decomposing formulas of the forms $[\alpha \parallel \beta]\,\varphi$ is resolved by extending the language and by adapting the Fischer-Ladner closure. In Sect. 4, the new concepts of threads, twines and neat models are introduced. And in Sect. 5, it is proved that the class of neat $\lhd$-deterministic frames satisfies the same formulas as the class of $\lhd$-deterministic frames. In Sect. 6, the strong finite model property is proved by piecewise filtration.

2 Propositional Dynamic Logic with Separating Parallel Composition (PPDL)

Let Π_0 be a countable set of atomic programs (denoted by $a, b \ldots$) and Φ_0 a countable set of propositional variables (denoted by $p, q \ldots$). The sets Π and Φ of programs and formulas are defined by:

$$\alpha, \beta := a \mid (\alpha\,;\beta) \mid (\alpha \cup \beta) \mid \varphi? \mid \alpha^* \mid (\alpha \parallel \beta)$$
$$\varphi := p \mid \bot \mid \neg\varphi \mid [\alpha]\,\varphi$$

The negation construct is an involution: by definition, $\neg\neg\varphi = \varphi$. Parentheses may be omitted for clarity, but they are taken into account when counting occurrences of symbols. We write $|\alpha|$ and $|\varphi|$ for the number of occurrences of symbols in the program α and the formula φ, respectively. We define the abbreviations $\top \doteq \neg\bot$ and $\langle\alpha\rangle\varphi \doteq \neg\,[\alpha]\,\neg\varphi$. The missing usual (additive) Boolean operators can be defined too, starting with $\varphi \rightarrow \psi \doteq [\varphi?]\,\psi$. Additionally, a multiplicative conjunction related to BBI [17] may be defined as $\varphi * \psi \doteq \langle\varphi? \parallel \psi?\rangle\top$.

A frame is a tuple $(W, R, \lhd)$ where W is a non-empty set of states (denoted by $w, x, y \ldots$), R is a function associating a binary relation over W to each atomic program and $\lhd$ is a ternary relation over W. Intuitively, $x\ R(a)\ y$ means that the program a can be executed in state x, reaching state y. Similarly, $x \lhd (y, z)$ means that x can be split into the substates y and z or equivalently that y and z can be merged to obtain x. When the merging of states is functional, the frame is said to be $\lhd$-deterministic. This is a common restriction, for instance in separation logics, expressing the fact that the parts determine the whole. Formally, a frame is $\lhd$-deterministic iff for all $x, y, w_1, w_2 \in W$, if $x \lhd (w_1, w_2)$ and $y \lhd (w_1, w_2)$ then $x = y$. The class of $\lhd$-deterministic frames is denoted by $\mathcal{C}_{\lhd\text{-det}}$.

A model is a tuple $(W, R, \lhd, V)$ where $(W, R, \lhd)$ is a frame and V is a function associating a subset of W to each propositional variable. A model is $\lhd$-deterministic iff its frame is $\lhd$-deterministic. The forcing relation $\vDash$ is defined by parallel induction along with the extension of R to all programs:

$$
\begin{array}{ll}
\mathcal{M}, x \vDash p & \text{iff } x \in V(p) \\
\mathcal{M}, x \vDash \bot & \text{never} \\
\mathcal{M}, x \vDash \neg\varphi & \text{iff } \mathcal{M}, x \nvDash \varphi \\
\mathcal{M}, x \vDash [\alpha]\,\varphi & \text{iff } \forall y \in W,\ \text{if } x\ R(\alpha)\ y \text{ then } \mathcal{M}, y \vDash \varphi \\
x\ R(\alpha\,;\beta)\ y & \text{iff } \exists z \in W,\ x\ R(\alpha)\ z \text{ and } z\ R(\beta)\ y \\
x\ R(\alpha \cup \beta)\ y & \text{iff } x\ R(\alpha)\ y \text{ or } x\ R(\beta)\ y \\
x\ R(\varphi?)\ y & \text{iff } x = y \text{ and } \mathcal{M}, x \vDash \varphi \\
x\ R(\alpha^*)\ y & \text{iff } x\ R(\alpha)^*\ y \\
 & \qquad \text{where } R(\alpha)^* \text{ is the reflexive and transitive closure of } R(\alpha) \\
x\ R(\alpha \parallel \beta)\ y & \text{iff } \exists w_1, w_2, w_3, w_4 \in W, \\
 & \qquad x \lhd (w_1, w_2), w_1\ R(\alpha)\ w_3, w_2\ R(\beta)\ w_4 \text{ and } y \lhd (w_3, w_4)
\end{array}
$$

Given a class $\mathcal{C}$ of frames, a formula φ is *satisfiable in* $\mathcal{C}$ iff there exists a model $\mathcal{M} = (W, R, \lhd, V)$ and a state $w \in W$ such that $(W, R, \lhd) \in \mathcal{C}$ and $\mathcal{M}, w \vDash \varphi$. The satisfiability problem of PPDL over a class $\mathcal{C}$ of frames is the decision problem determining whether a PPDL formula is satisfiable in $\mathcal{C}$.

3 Fischer-Ladner Closure

In [10], Ficher and Ladner proved the strong finite model property of PDL by means of the filtration by a set of formulas called the Fischer-Ladner closure.

To cope with nondeterministic choice and iteration, the original Fischer-Ladner closure extends PDL's language with new propositional variables. In the case of PPDL, a more involved extension of the language is needed to cope with parallel composition of programs. We first introduce this extension before defining the Fischer-Ladner closure adapted to PPDL.

3.1 Placeholders and Marking Functions

In order to decompose formulas of the form $[\alpha \parallel \beta]\,\varphi$ into subformulas, the language is extended with new atomic formulas called placeholders and parallel composition symbols are distinguished by added indices. Using the same sets Φ_0 and Π_0 of propositional variables and atomic programs, the sets Π_{PH}, Φ_{pure} and Φ_{PH} of *annotated programs*, *pure formulas* and *annotated formulas* respectively, are defined by parallel induction as follows:

$$\begin{aligned}\alpha, \beta &:= a \mid (\alpha \,;\, \beta) \mid (\alpha \cup \beta) \mid \varphi? \mid \alpha^* \mid (\alpha \parallel_i \beta)\\ \varphi &:= p \mid \bot \mid \neg\varphi \mid [\alpha]\varphi\\ \psi &:= \varphi \mid (i, j) \mid \neg\psi \mid [\alpha]\psi\end{aligned}$$

where i ranges over $\mathbb{N}$ and j over $\{1, 2\}$. Moreover, for all $i \in \mathbb{N}$, there must be at most one occurrence of $\parallel_i$ in any annotated program, any pure formula and any annotated formula. The integers below the parallel composition symbols are called *indices*. Formulas of the form (i, j) are called *placeholders*.

To interpret the annotated formulas, if placeholders were simply considered as new propositional variables, it would be impossible to ensure that whenever $w \lhd (x, y)$ and $\mathcal{M}, w \vDash [\alpha \parallel_i \beta]\,\varphi$ then $\mathcal{M}, x \vDash [\alpha]\,(i, 1)$ and $\mathcal{M}, y \vDash [\beta]\,(i, 2)$. Therefore we add flexibility in the interpretation of placeholders by adding *marking functions* which are functions from placeholders to subset of W. The set of all such functions is denoted by B_W. The *empty marking function* $m_W^{\varnothing} \in B_W$ binds the empty set to all placeholders. The 4-ary forcing relation $\vDash_{\text{F}}$ is defined on all models $\mathcal{M} = (W, R, \lhd, V)$, all $w \in W$, all $m \in B_W$ and all $\varphi \in \Phi_{PH}$ by parallel induction along with the extension of R to all annotated programs, in a similar way than for PPDL except:

$$\begin{aligned}&\mathcal{M}, x, m \vDash_{\text{F}} (i, j) && \text{iff } x \in m(i, j)\\ &x\ R(\varphi?)\ y && \text{iff } x = y \text{ and } \mathcal{M}, x, m_W^{\varnothing} \vDash_{\text{F}} \varphi\\ &x\ R(\alpha \parallel_i \beta)\ y && \text{iff } \exists w_1, w_2, w_3, w_4,\\ &&& \quad x \lhd (w_1, w_2), w_1\ R(\alpha)\ w_3, w_2\ R(\beta)\ w_4 \text{ and } y \lhd (w_3, w_4)\end{aligned}$$

There exists a forgetful surjection $\overline{\cdot} : \Phi_{\text{pure}} \longrightarrow \Phi$ associating to each pure formula φ the formula $\overline{\varphi}$ obtained by removing all indices in φ. Thanks to the following lemma, we will consider satisfiability of pure formulas instead of satisfiability of PPDL formulas.

Lemma 1. *For all model $\mathcal{M} = (W, R, \lhd, V)$, all $\varphi \in \Phi_{pure}$ and all $w \in W$, $\mathcal{M}, w, m_W^{\varnothing} \vDash_F \varphi$ iff $\mathcal{M}, w \vDash \overline{\varphi}$.*

$$\frac{(\mu,\ \varphi)}{(\mu,\ \neg\varphi)} \qquad \frac{(\mu,\ [a]\,\varphi)}{(\mu,\ \varphi)}$$

$$\frac{(\mu,\ [\alpha\,;\beta]\,\varphi)}{(\mu,\ [\alpha]\,[\beta]\,\varphi)} \qquad \frac{(\mu,\ [\alpha\cup\beta]\,\varphi)}{(\mu,\ [\alpha]\,\varphi) \quad (\mu,\ [\beta]\,\varphi)}$$

$$\frac{(\mu,\ [\varphi?]\,\psi)}{(\mu,\ \varphi) \quad (\mu,\ \psi)} \qquad \frac{(\mu,\ [\alpha^*]\,\varphi)}{(\mu,\ [\alpha]\,[\alpha^*]\,\varphi) \quad (\mu,\ \varphi)}$$

$$\frac{(\mu,\ [\alpha\parallel_i\beta]\,\varphi)}{(\mu.L,\ [\alpha]\,(i,1)) \quad (\mu.R,\ [\beta]\,(i,2)) \quad (\mu,\ \varphi)}$$

Fig. 1. Fischer-Ladner closure calculus

3.2 Fischer-Ladner Closure

The Fischer-Ladner closure is a decomposition of any PDL formula into a set containing sufficiently many subformulas for the filtration. In the case of PPDL, we need to keep track of the level of separation (called *depth*) of each subformula. Hence we consider *localized formulas*. A *location* is a word on the alphabet $\{L, R\}$, the empty word being denoted by ϵ. A localized formula is a pair $(\mu,\ \varphi)$ composed of a location μ and a formula φ.

Then, given a localized formula $(\mu,\ \varphi)$ over Φ_0 and Π_0, we construct the *closure* $\mathrm{Cl}(\mu,\ \varphi)$ *of* $(\mu,\ \varphi)$ by applying the rules in Fig. 1. In the remainder of this paper we will be mainly interested in closure of localized formulas of the form $(\epsilon,\ \varphi_0)$ where φ_0 is a pure formula. For all pure formula $\varphi_0 \in \Phi_{\mathrm{pure}}$, we define the abbreviations $\mathrm{FL}(\varphi_0) = \mathrm{Cl}(\epsilon,\ \varphi_0)$ and $\mathrm{SP}(\varphi_0) = \{\alpha \mid \exists\mu, \exists\varphi, (\mu,\ \langle\alpha\rangle\varphi) \in \mathrm{FL}(\varphi_0)\}$. The cardinality of $\mathrm{FL}(\varphi_0)$ is denoted by N_{φ_0}. The proof from [10] can be easily adapted to prove the following lemma:

Lemma 2. *N_{φ_0} is linear in $|\varphi_0|$.*

4 Threads, Twines and Neat Models

In this section, new concepts about PPDL's models are introduced. These concepts allow us to restrict the class of models to consider for satisfiability. In the next section, we prove that any formula satisfiable in the class of $\lhd$-deterministic models is satisfied in a model with these additional properties. Firstly, to bound the depth of separation of states, we introduce the notion of hierarchical models. This directly corresponds to locations of formulas from the previous section.

Definition 1. *Given a model $\mathcal{M} = (W, R, \lhd, V)$, a function $\lambda : W \longrightarrow \{L, R\}^*$ is a hierarchy function for $\mathcal{M}$ iff*

$$\forall x, y, z \in W, \qquad x \lhd (y, z) \Rightarrow \lambda(y) = \lambda(x).L \text{ and } \lambda(z) = \lambda(x).R \quad (1)$$

$$\forall x, y \in W, \forall\alpha \in \Pi_{PH}, \qquad x\ R(\alpha)\ y \Rightarrow \lambda(x) = \lambda(y) \quad (2)$$

$\lambda(x)$ is called the depth *of x. A model for which there exists a hierarchy function is a* hierarchical model.

Secondly, in order to restrict the number of states at each level of separation, the notion of reachability is extended. Given a $\lhd$-deterministic model $\mathcal{M} = (W, R, \lhd, V)$, consider the reachability relation $R^{\exists} = \cup_{\alpha \in \Pi_{PH}} R(\alpha)$. This relation is obviously reflexive. Hence its symmetric and transitive closure, denoted by $\sim$, is an equivalence relation. The equivalence classes of W by $\sim$ are called *threads* and $\sim$ the thread relation. Notice that if $\mathcal{M}$ is hierarchical, all states in any thread T have the same depth, noted $\lambda(T)$. To strengthen the link between threads and depth, threads are grouped into pairs, each thread of a pair corresponding to one side of the separations. These pairs of threads are called twines and are formally defined as follows:

Definition 2. *A* twine *is a pair* (T_L, T_R) *of threads such that for all* $x, y, z \in W$ *if* $x \lhd (y, z)$ *then* $y \notin T_R$*,* $z \notin T_L$ *and* $y \in T_L \Leftrightarrow z \in T_R$*.*

In the remainder of this paper, a twine (T_1, T_2) is identified with the set $T_1 \cup T_2$. Obviously, if a thread T is such that for all $(x, y, z) \in \lhd$, $y \notin T$ and $z \notin T$, then for any thread T' having the same property, (T, T') is a twine. Such a thread is called an *isolated thread*. It can be easily proved that if (T_1, T_2) and (T_1, T_3) are twines, then either T_1, T_2 and T_3 are isolated or $T_2 = T_3$. We can now define the notion of neat models.

Definition 3. *A model* $\mathcal{M} = (W, R, \lhd, V)$ *is* neat *if it satisfies all the following conditions:*

1. *For any thread* T_1 *there exists a thread* T_2 *such that* (T_1, T_2) *or* (T_2, T_1) *is a twine;*
2. *There is exactly one isolated thread* T_0*;*
3. *There exists a hierarchy function* λ *for* $\mathcal{M}$ *such that* $\lambda(T_0) = \epsilon$*.*

5 Neat Model Property

In this section we will prove that whenever a pure formula is satisfiable in $\mathcal{C}_{\lhd\text{-det}}$, it is satisfiable in a $\lhd$-deterministic neat model. Supposing the pure formula φ_0 is satisfiable, the proof proceeds as follows:

- by Lemma 3 below, there exists a countable model $\mathcal{M}_{\mathfrak{B}}$ satisfying φ_0;
- in Sect. 5.1, $\mathcal{M}_{\mathfrak{B}}$ is unraveled into $\mathcal{M}_{\mathfrak{U}}$.
- in Sect. 5.2, unreachable states from $\mathcal{M}_{\mathfrak{U}}$ are pruned to obtain $\mathcal{M}_{\mathfrak{N}}$ and $\mathcal{M}_{\mathfrak{N}}$ is proved to be a $\lhd$-deterministic neat model satisfying φ_0.

Lemma 3. *For any satisfiable pure formula* φ_0*, there exists a countable model satisfying* φ_0*.*

Proof. By a proof similar to Corollary 6.3 in [3]. □

5.1 Unraveling

Let $\mathcal{M}_{\mathfrak{B}} = (W_{\mathfrak{B}}, R_{\mathfrak{B}}, \lhd_{\mathfrak{B}}, V_{\mathfrak{B}})$ be a countable $\lhd$-deterministic model satisfying a formula φ_0 at x'_0. We will construct the *unraveling* of $\mathcal{M}_{\mathfrak{B}}$ at x'_0. The following method is an adaptation of the well-known unraveling method (see [5] for instance). The key difference is that the resulting model is not a tree-like model.

Let W_∞ be a countably infinite set. For all $k \in \mathbb{N}$ we will construct the tuple $U_k = (\mathcal{M}_k, h_k)$ such that $\mathcal{M}_k = (W_k, R_k, \lhd_k, V_k)$ is a model with $W_k \subseteq W_\infty$ and h_k is a homomorphism from W_k to $W_{\mathfrak{B}}$, thus preserving valuation. The initial tuple U_0 is such that $W_0 = \{x_0\}$ for some $x_0 \in W_\infty$, $R_0(a) = \emptyset$ for all $a \in \Pi_0$, $\lhd_0 = \emptyset$ and $h_0(x_0) = x'_0$. Then for all $k \in \mathbb{N}$, U_{k+1} is constructed from U_k by fixing one of the following defects for some $v, w_1, w_2 \in W_k$, $a \in \Pi_0$ and $w', w'_1, w'_2 \in W_{\mathfrak{B}}$:

Successor Defect (v, a, w')**.** If $h_k(v)\ R_{\mathfrak{B}}(a)\ w'$ but there is no $w \in W_k$ such that $h_k(w) = w'$ and $v\ R_k(a)\ w$, then U_{k+1} is obtained from U_k by adding a new state $w \in W_\infty \setminus W_k$ such that $h_{k+1}(w) = w'$ and $v\ R_{k+1}(a)\ w$,

Split Defect (v, w'_1, w'_2)**.** If $h_k(v) \lhd_{\mathfrak{B}} (w'_1, w'_2)$ but there are no $w_1, w_2 \in W_k$ such that $h_k(w_1) = w'_1$, $h_k(w_2) = w'_2$ and $v \lhd_k (w_1, w_2)$, then U_{k+1} is obtained from U_k by adding two new states $w_1, w_2 \in W_\infty \setminus W_k$ such that $h_{k+1}(w_1) = w'_1$, $h_{k+1}(w_2) = w'_2$ and $v \lhd_{k+1} (w_1, w_2)$.

Merge Defect (w', w_1, w_2)**.** If $w' \lhd_{\mathfrak{B}} (h_k(w_1), h_k(w_2))$ but there is no $w \in W_k$ such that $h_k(w) = w'$ and $w \lhd_k (w_1, w_2)$, then U_{k+1} is obtained from U_k by adding a new state $w \in W_\infty \setminus W_k$ such that $h_{k+1}(w) = w'$ and $w \lhd_{k+1} (w_1, w_2)$.

Since W_∞, Π_0 and $W_{\mathfrak{B}}$ are countable sets, there is a sequence $\delta_0, \delta_1, \ldots$ of possible defects such that each possible defect appears infinitely often. We enforce that for all $k \in \mathbb{N}$, either δ_k is a defect for U_k fixed in U_{k+1} or δ_k is not a defect for U_k and $U_{k+1} = U_k$. The unraveling $\mathcal{M}_{\mathfrak{U}} = (W_{\mathfrak{U}}, R_{\mathfrak{U}}, \lhd_{\mathfrak{U}}, V_{\mathfrak{U}})$ of $\mathcal{M}_{\mathfrak{B}}$ at x'_0 is the union of $\mathcal{M}_k$ for all $k \in \mathbb{N}$.

Proposition 1. *$\mathcal{M}_{\mathfrak{U}}$ is a $\lhd$-deterministic model satisfying φ_0.*

To prove Proposition 1, we adapt the bounded morphism definition to PPDL and prove Lemma 4.

Definition 4. *Given two $\lhd$-deterministic models $\mathcal{M} = (W, R, \lhd, V)$ and $\mathcal{M}' = (W', R', \lhd', V')$, a mapping $h : \mathcal{M} \longrightarrow \mathcal{M}'$ is* a bounded morphism *iff it satisfies the following conditions for all $v, w, w_1, w_2 \in W$, $w', w'_1, w'_2 \in W'$ and $a \in \Pi_0$:*

$$w \text{ and } h(w) \text{ satisfy the same propositional variables} \tag{3}$$

$$v\ R(a)\ w \Rightarrow h(v)\ R'(a)\ h(w) \tag{4}$$

$$h(v)\ R'(a)\ w' \Rightarrow \exists w,\ h(w) = w' \text{ and } v\ R(a)\ w \tag{5}$$

$$w \lhd (w_1, w_2) \Rightarrow h(w) \lhd' (h(w_1), h(w_2)) \tag{6}$$

$$h(w) \lhd' (w'_1, w'_2) \Rightarrow \exists w_1, w_2,\ \begin{cases} h(w_1) = w'_1, h(w_2) = w'_2 \\ \qquad \text{and } w \lhd (w_1, w_2) \end{cases} \tag{7}$$

$$w' \lhd' (h(w_1), h(w_2)) \Rightarrow \exists w,\ h(w) = w' \text{ and } w \lhd (w_1, w_2) \tag{8}$$

Lemma 4. *If h is a bounded morphism from $\mathcal{M}$ to $\mathcal{M}'$, then for all $w \in W$ and $\varphi \in \Phi$, $\mathcal{M}, w \vDash \varphi$ iff $\mathcal{M}', h(w) \vDash \varphi$.*

Considering the functions $(h_k)_{k\in\mathbb{N}}$ as subsets of $W_\mathfrak{U} \times W_\mathfrak{B}$, we define h as their union. We prove that h is a bounded morphism, the successor, split and merge defects ensuring conditions (5), (7) and (8) respectively. Finally, since bounded morphisms preserve $\lhd$-determinism, Proposition 1 is proved. Despite $\mathcal{M}_\mathfrak{U}$ not being tree-like, it has the following form of acyclicity:

Proposition 2. *For all $x, y \in W_\mathfrak{U}$ and all $\alpha, \beta \in \Pi_{PH}$, if $x\ R(\alpha)\ y$ and $y\ R(\beta)\ x$ then $x = y$.*

5.2 Pruning

In this section, we remove unreachable states from $\mathcal{M}_\mathfrak{U}$ and prove that the resulting model is neat. The method consists in identifying reachable threads and relies on the fact that new reachable threads are added only by split defects. We use a function r associating to each state $x \in W_\mathfrak{U}$ either the first state of x's thread if this thread is reachable or the special value Out otherwise. The function $r : W_\mathfrak{U} \longrightarrow W_\mathfrak{U} \cup \{\mathsf{Out}\}$ is formally defined by induction on the construction of $\mathcal{M}_\mathfrak{U}$ as follows:

0. Initially, $r(x_0) = x_0$;
1. When fixing a successor defect (w, a, v) by adding w', $r(w') = r(w)$;
2. When fixing a split defect (w, v_1, v_2) by adding w_1 and w_2, if $r(w) \neq \mathsf{Out}$ then $r(w_1) = w_1$ and $r(w_2) = w_2$, otherwise $r(w_1) = r(w_2) = \mathsf{Out}$;
3. When fixing a merge defect $\delta_k = (v, w_1, w_2)$ by adding w, if there exists $w' \in W_k$ such that $w' \lhd_k (r(w_1), r(w_2))$ then $r(w) = r(w')$, otherwise $r(w) = \mathsf{Out}$.

The function r is well-defined because $\mathcal{M}_\mathfrak{U}$ is $\lhd$-deterministic. Then, the model $\mathcal{M}_\mathfrak{N} = (W_\mathfrak{N}, R_\mathfrak{N}, \lhd_\mathfrak{N}, V_\mathfrak{N})$ is defined as the reduction of $\mathcal{M}_\mathfrak{U}$ to the worlds x for which $r(x) \neq \mathsf{Out}$. The following proposition can easily be proved:

Proposition 3. *$\mathcal{M}_\mathfrak{N}$ is a $\lhd$-deterministic model satisfying φ_0 at x_0.*

It remains to prove that $\mathcal{M}_\mathfrak{N}$ is neat. Let $\sim_\mathfrak{N}$ be the thread relation of $\mathcal{M}_\mathfrak{N}$. The proof of Proposition 4 relies on the following two lemmas:

Lemma 5. *For all $x, y \in W_\mathfrak{N}$, $r(x) = r(y)$ iff $x \sim_\mathfrak{N} y$.*

Lemma 6. *If $z \lhd_\mathfrak{N} (x, y)$ then there exists $z' \in W_\mathfrak{N}$ such that $z' \lhd_\mathfrak{N} (r(x), r(y))$ and $(z', r(x), r(y))$ has been added to $\lhd_\mathfrak{U}$ by a split defect.*

Proposition 4. *$\mathcal{M}_\mathfrak{N}$ is neat.*

Proof sketch. For the first two conditions of Definition 3, we prove the corresponding two properties using Lemma 6:

1. For all $x_1, x_2, y_1, y_2, z_1, z_2 \in W_\mathfrak{N}$, if $z_1 \lhd_\mathfrak{N} (x_1, y_1)$ and $z_2 \lhd_\mathfrak{N} (x_2, y_2)$ then $r(x_1) = r(x_2) \Leftrightarrow r(y_1) = r(y_2)$.

2. x_0 is the only $x \in W_{\mathfrak{N}}$ such that $r(x) = x$ and for all $(w, y, z) \in \lhd_{\mathfrak{N}}$, $r(y) \neq x$ and $r(z) \neq x$.

For the last condition of Definition 3, the hierarchy function λ is constructed such that:

- $\lambda(x_0) = \epsilon$;
- for any split defect (w, v_1, v_2) adding w_1 and w_2 to $W_{\mathfrak{U}}$, if $r(w) \neq \mathsf{Out}$ then $\lambda(w_1) = \lambda(w).L$ and $\lambda(w_2) = \lambda(w).R$;
- for all x, $\lambda(x) = \lambda(r(x))$. □

6 Piecewise Filtration

In this section, we prove the following proposition:

Proposition 5. *Whenever a formula $\varphi \in \Phi$ is satisfiable in a $\lhd$-deterministic neat model, it is satisfiable in a $\lhd$-deterministic finite model $\mathcal{M} = (W, R, \lhd, V)$ in which the cardinality of W is bounded by an exponential in the number of symbols in φ.*

Suppose $\mathcal{M}_{\mathfrak{N}} = (W_{\mathfrak{N}}, R_{\mathfrak{N}}, \lhd_{\mathfrak{N}}, V_{\mathfrak{N}})$ is neat and $\mathcal{M}_{\mathfrak{N}}, x_0, m_W^{\varnothing} \vDash_{\mathrm{F}} \varphi_0$ for some $x_0 \in W_{\mathfrak{N}}$ and $\varphi_0 \in \Phi_{\mathrm{pure}}$. Furthermore, we suppose λ is a hierarchical function for $\mathcal{M}_{\mathfrak{N}}$ such that $\lambda(x_0) = \epsilon$. The model $\mathcal{M}_{\mathfrak{F}}$ satisfying Proposition 5 is inductively constructed from $\mathcal{M}_{\mathfrak{N}}$. At the initial step, the filtration of the thread containing x_0 is added to $\mathcal{M}_{\mathfrak{F}}$. At the inductive steps, for each pair of states in $\mathcal{M}_{\mathfrak{F}}$ which must be connected by a parallel program, the filtration of a twine of $\mathcal{M}_{\mathfrak{N}}$ corresponding to this parallel program is added to $\mathcal{M}_{\mathfrak{F}}$.

In order to preserve the $\lhd$-determinism of $\mathcal{M}_{\mathfrak{N}}$ during the filtration, we need to distinguish for any filtered twine, the forward (split) decomposition from the backward (merge) one. For that matter, placeholders are duplicated and the special pair $\{(0, 1), (0, 2)\}$ of placeholders is used to mark the forward decomposition. Formally, for any formula $\varphi \in \Phi_{PH}$ and any $k \in \mathbb{N}$, let $f_{\mathrm{dup}}(k, \varphi)$ be the formula obtained from φ by replacing each occurrence of (i, j) in φ by $(2i + k, j)$, for all $i \in \mathbb{N}$ and $j \in \{1, 2\}$. We define the sets

$$\begin{aligned} \mathrm{FL}^+(\varphi_0) = & \{(\mu,\ f_{\mathrm{dup}}(k, \varphi)) \mid k \in \{1, 2\}, (\mu,\ \varphi) \in \mathrm{FL}(\varphi_0)\} \cup \\ & \{((\mu,\ (0, j)), (\mu,\ \neg(0, j)) \mid j \in \{1, 2\} \text{ and } \exists \varphi, (\mu,\ \varphi) \in \mathrm{FL}(\varphi_0)\} \\ \mathrm{SF}^+(\varphi_0) = & \left\{\varphi \mid \exists \mu,\ (\mu,\ \varphi) \in \mathrm{FL}^+(\varphi_0)\right\} \end{aligned}$$

The filtrations are done using the $\equiv_m$ equivalence relations over $W_{\mathfrak{N}}$, defined for any marking function $m \in B_{W_{\mathfrak{N}}}$ such that $x \equiv_m y$ iff $\lambda(x) = \lambda(y)$ and for all $(\mu, \varphi) \in \mathrm{FL}^+(\varphi_0)$ if $\mu = \lambda(x)$ then $\mathcal{M}_{\mathfrak{N}}, x, m \vDash_{\mathrm{F}} \varphi \Leftrightarrow \mathcal{M}_{\mathfrak{N}}, y, m \vDash_{\mathrm{F}} \varphi$. The functions Ω and Ψ are defined for all $X \subseteq W_{\mathfrak{N}}$ and $m \in B_{W_{\mathfrak{N}}}$ by:

$$\begin{aligned} \Omega(X, m) &= \left\{Y \cap X \mid Y \in W_{\mathfrak{N}}/\!\equiv_m\right\} \\ \Psi(X, m) &= \left\{\varphi \mid \exists x \in W_{\mathfrak{N}},\ (\lambda(x), \varphi) \in \mathrm{FL}^+(\varphi_0) \text{ and } \mathcal{M}_{\mathfrak{N}}, x, m \vDash_{\mathrm{F}} \varphi\right\} \end{aligned}$$

Finally, the set PF references all the parallel program links for which we may have to add the filtration of a twine. Formally, PF is the greatest subset of $\mathbb{N} \times \mathcal{P}\left(\mathrm{SF}^{+}(\varphi_0)\right) \times \mathrm{SP}(\varphi_0) \times \mathcal{P}\left(\mathrm{SF}^{+}(\varphi_0)\right)$ such that for all $(k, F, \alpha, G) \in \mathrm{PF}$, α is of the form $(\alpha_1 \parallel_i \alpha_2)$ and there exists $\mu \in \{L, R\}^*$ and $\varphi \in \Phi_{PH}$ such that $(\mu, \langle\alpha\rangle\varphi) \in \mathrm{FL}^{+}(\varphi_0)$ and for all $\psi \in F \cup G$, $(\mu, \psi) \in \mathrm{FL}^{+}(\varphi_0)$. Since $\mathcal{P}\left(\mathrm{SF}^{+}(\varphi_0)\right)$ and $\mathrm{SP}(\varphi_0)$ are finite, there exists a total order over PF with a least element and such that $(k, F, \alpha, G) < (k', F', \alpha', G')$ implies $k \leq k'$. This order determines a bijective function from $\mathbb{N}$ to PF. Moreover, if (k, F, α, G) is the n^{th} tuple in PF then $k \leq n$.

Now we inductively construct the models $\mathcal{M}_n = (W_n, R_n, \lhd_n, V_n)$ for $n \in \mathbb{N}$, where $W_n \subseteq \mathbb{N} \times \mathcal{P}(W_{\mathfrak{N}}) \times B_{W_{\mathfrak{N}}}$. The following invariants hold for all $n \in \mathbb{N}$:

- for all $(k, X, m) \in W_n$, all $\varphi \in \Psi(X, m)$ and all $x \in X$, $\mathcal{M}_{\mathfrak{N}}, x, m \vDash_{\mathrm{F}} \varphi$;
- for all $(k, X, m), (k', Y, m') \in W_n$, if $k = k'$ then $m = m'$ and for all $x \in X$ and all $y \in Y$, x and y belong to the same twine and if $x \equiv_m y$ then $X = Y$.

Initial step. Let T_0 be the thread in $\mathcal{M}_{\mathfrak{N}}$ containing x_0. We set:

$$\begin{aligned}
W_0 &= \left\{(0, X, m^{\varnothing}_{W_{\mathfrak{N}}}) \;\middle|\; X \in \Omega(T_0, m^{\varnothing}_{W_{\mathfrak{N}}})\right\} \\
R_0(a) &= \{((k, X, m), (k', X', m')) \in W_0 \times W_0 | \\
&\qquad k = k' \text{ and } \exists x \in X, \exists x' \in X',\ x\ R_{\mathfrak{N}}(a)\ x'\} \\
\lhd_0 &= \emptyset \\
V_0(p) &= \{(k, X, m) \in W_0 \mid p \in \Psi(X, m)\}
\end{aligned}$$

If $\mathrm{PF} = \emptyset$ then $\mathcal{M}_n = \mathcal{M}_0$ for all $n \in \mathbb{N}$. Otherwise the following inductive step is applied.

Inductive step. Suppose $\mathcal{M}_n$ has already been defined and let $(k, F, \alpha_1 \parallel_i \alpha_2, G)$ be the n^{th} tuple in PF. If for all $X, Y \subseteq W_{\mathfrak{N}}$ and all $m \in B_{W_{\mathfrak{N}}}$, one of the following conditions is not satisfied

$$\Psi(X, m) = F \text{ and } \Psi(Y, m) = G \tag{9}$$

$$(k, X, m) \in W_n \text{ and } (k, Y, m) \in W_n \tag{10}$$

$$\exists x \in X, \exists y \in Y,\ x\ R_{\mathfrak{N}}(\alpha_1 \parallel_i \alpha_2)\ y \tag{11}$$

then $\mathcal{M}_{n+1} = \mathcal{M}_n$. Otherwise, by the invariants, there is exactly one tuple (X, Y, m) satisfying (9) and (10). By condition (11), there exists $x \in X$, $y \in Y$ and $w_1, w_2, w_3, w_4 \in W_{\mathfrak{N}}$ such that $x \lhd_{\mathfrak{N}} (w_1, w_2)$, $w_1\ R_{\mathfrak{N}}(\alpha_1)\ w_3$, $w_2\ R_{\mathfrak{N}}(\alpha_2)\ w_4$ and $y \lhd_{\mathfrak{N}} (w_3, w_4)$. The marking function m_{n+1} is defined such that

- $m_{n+1}(0, j) = \{w_j\}$;
- $m_{n+1}(i, j) = \;\; w \;\Big|\; \exists \beta_1, \beta_2.(\beta_1 \parallel_{\frac{i-1}{2}} \beta_2) \in \mathrm{SP}(\varphi_0)$ and $w_j\ R_{\mathfrak{N}}(\beta_j)\ w$ if i is odd;
- $m_{n+1}(i, j) = \;\; w \;\Big|\; \exists \beta_1, \beta_2, (\beta_1 \parallel_{\frac{i-2}{2}} \beta_2) \in \mathrm{SP}(\varphi_0)$ and $w_{j+2}\ R_{\mathfrak{N}}(\beta_j)\ w$ if i is even and positive.

Since $\mathcal{M}_{\mathfrak{N}}$ is neat, w_1, w_2, w_3 and w_4 belong to the same twine θ. For all $t \in 1..4$, there exists $X_t \in \Omega(\theta, m_{n+1})$ such that $x_t \in X_t$. $\mathcal{M}_{n+1}$ is defined by:

$$\begin{aligned} W_{n+1} &= W_n \cup \{(n+1, X, m_{n+1}) \mid X \in \Omega(\theta, m_{n+1})\} \\ R_{n+1}(a) &= \{((k,X,m),(k',X',m')) \in W_{n+1} \times W_{n+1} | \\ &\qquad k = k' \text{ and } \exists x \in X, \exists x' \in X',\ x\ R_{\mathfrak{N}}(a)\ x'\} \\ \lhd_{n+1} &= \lhd_n \cup \{((k,X,m),(n+1,X_1,m_{n+1}),(n+1,X_2,m_{n+1})), \\ &\qquad ((k,Y,m),(n+1,X_3,m_{n+1}),(n+1,X_4,m_{n+1}))\} \\ V_{n+1}(p) &= \{(k,X,m) \in W_{n+1} \mid p \in \Psi(X,m)\} \end{aligned}$$

Finally, $\mathcal{M}_{\mathfrak{F}} = (W_{\mathfrak{F}}, R_{\mathfrak{F}}, \lhd_{\mathfrak{F}}, V_{\mathfrak{F}})$ is defined as the union of $\mathcal{M}_n$ for all $n \in \mathbb{N}$ and we prove that $\mathcal{M}_{\mathfrak{F}}$ satisfies Proposition 5.

Proof sketch of Proposition 5. To prove that the cardinality of $W_{\mathfrak{F}}$ is bounded by an exponential in $|\varphi_0|$, we consider the graph whose vertices are sets of $W_{\mathfrak{F}}$'s states having the same first component and such that there is an edge from G to G' iff states in G' have been added to $\mathcal{M}_{\mathfrak{F}}$ to connect two states in G. We prove that each vertex contains a exponential number of states and that the graph is a tree with branching factor exponential in $|\varphi_0|$ and depth linear in $|\varphi_0|$. $\lhd$-determinism of $\mathcal{M}_{\mathfrak{F}}$ is ensured by the interpretation of the placeholders $(0,1)$ and $(0,2)$. Finally, to prove the truth lemma, we prove the following properties by simultaneous induction on $|\varphi|$ for 1 and on $|\alpha|$ for 2 and 3:

1. For all $(k,X,m) \in W_{\mathfrak{F}}$ and all formula φ such that $(\lambda(X), \varphi) \in \mathrm{FL}^+(\varphi_0)$, $\varphi \in \Psi(X,m) \Leftrightarrow \mathcal{M}_{\mathfrak{F}}, (k,X,m), m_{\mathfrak{F}} \vDash_{\mathrm{F}} \varphi$.
2. If $(k,X,m) \in W_{\mathfrak{F}}$, $x \in X$, θ_x is the twine of x, $Y \in \Omega(\theta_x, m)$, $y \in Y$, $(\lambda(x), \langle\alpha\rangle\varphi) \in \mathrm{FL}(\varphi_0)$ and $x\ R_{\mathfrak{N}}(\alpha)\ y$, then $(k,X,m)\ R_{\mathfrak{F}}(\alpha)\ (k,Y,m)$.
3. If $(k,X,m)\ R_{\mathfrak{F}}(\alpha)\ (k,Y,m)$ and $[\alpha]\varphi \in \Psi(X,m)$, then $\varphi \in \Psi(Y,m)$. □

Finally, since the model checking problem for PPDL is obviously polynomial in the number of states of the model, we deduce a complexity upper bound:

Proposition 6. *The satisfiability problem of PPDL interpreted over $\lhd$-deterministic frames is in NEXPTIME.*

7 Conclusion

In this paper, we prove that PPDL interpreted over the class $\mathcal{C}_{\lhd\text{-det}}$ of $\lhd$-deterministic frames has a strong finite model property and that the satisfiability problem of this logic is in NEXPTIME. This results rely on the neat model property introduced in the paper and are obtained by a piecewise filtration using an adaptation of the Fischer-Ladner closure. Because formulas with parallel compositions cannot be properly decomposed into subformulas, the language is extended with indices and placeholders. We hope these new concepts will be useful in future works. We briefly list some possibilities. First, a tight complexity result for PPDL over $\mathcal{C}_{\lhd\text{-det}}$ remains to be found. Secondly, the complexity of PPDL interpreted over the class of neat frames could be studied. Finally, since no semantic equivalents of the multiplicative implication of BBI can be defined in PPDL over $\mathcal{C}_{\lhd\text{-det}}$, it could explicitly be added to the language.

References

1. Abrahamson, K.R.: Modal logic of concurrent nondeterministic programs. In: Kahn, G. (ed.) Semantics of Concurrent Computation. LNCS, vol. 70, pp. 21–33. Springer, Heidelberg (1979)
2. Balbiani, P., Boudou, J.: Iteration-free PDL with storing, recovering and parallel composition: a complete axiomatization. J. Logic Comput. (2015, to appear)
3. Balbiani, P., Tinchev, T.: Definability and computability for PRSPDL. In: Advances in Modal Logic, pp. 16–33. College Publications (2014)
4. Benevides, M.R.F., de Freitas, R.P., Viana, J.P.: Propositional dynamic logic with storing, recovering and parallel composition. ENTCS **269**, 95–107 (2011)
5. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge Tracts in Theoretical Computer Science, vol. 53. Cambridge University Press, Cambridge (2001)
6. Cardelli, L., Gordon, A.D.: Anytime, anywhere: modal logics for mobile ambients. In: POPL, pp. 365–377. ACM (2000)
7. Collinson, M., Pym, D.J.: Algebra and logic for resource-based systems modelling. Math. Struct. Comput. Sci. **19**(5), 959–1027 (2009)
8. Demri, S., Deters, M.: Separation logics and modalities: a survey. J. Appl. Non Class. Logics **25**(1), 50–99 (2015)
9. van Ditmarsch, H., van der Hoek, W., Kooi, B.P.: Dynamic Epistemic Logic, vol. 337. Springer Science and Business Media, Heidelberg (2007)
10. Fischer, M.J., Ladner, R.E.: Propositional dynamic logic of regular programs. J. Comput. Syst. Sci. **18**(2), 194–211 (1979)
11. Harel, D.: Recurring dominoes: making the highly undecidable highly understandable (preliminary report). In: Budach, L. (ed.) Fundamentals of Computation Theory. LNCS, vol. 158, pp. 177–194. Springer, Heidelberg (1983)
12. Lange, M., Lutz, C.: 2-exptime lower bounds for propositional dynamic logics with intersection. J. Symb. Log. **70**(4), 1072–1086 (2005)
13. Larchey-Wendling, D., Galmiche, D.: The undecidability of boolean BI through phase semantics. In: LICS, pp. 140–149. IEEE Computer Society (2010)
14. Marx, M., Pólos, L., Masuch, M.: Arrow Logic and Multi-modal Logic. CSLI Publications, Stanford (1996)
15. Mayer, A.J., Stockmeyer, L.J.: The complexity of PDL with interleaving. Theor. Comput. Sci. **161**(1–2), 109–122 (1996)
16. Peleg, D.: Concurrent dynamic logic. J. ACM **34**(2), 450–479 (1987)
17. Pym, D.J.: The semantics and proof theory of the logic of bunched implications, Applied Logic Series, vol. 26. Kluwer Academic Publishers (2002)
18. Reynolds, J.C.: Separation logic: A logic for shared mutable data structures. In: LICS. pp. 55–74. IEEE Computer Society (2002)