Axiomatization and Computability of a Variant of Iteration-Free PDL with Fork

Philippe Balbiani(✉) and Joseph Boudou

Institut de recherche en informatique de Toulouse,
CNRS — Université de Toulouse, 118 route de Narbonne,
31062 Toulouse Cedex 9, France
Philippe.Balbiani@irit.fr

Abstract. We devote this paper to the axiomatization and the computability of PDL_0^{Δ}—a variant of iteration-free PDL with fork.

Keywords: Iteration-free PDL · Fork · Axiomatization
Computability

1 Introduction

Propositional dynamic logic (PDL) is an applied non-classical logic designed for reasoning about the behaviour of programs [10]. The definition of its syntax is based on the idea of associating with each program α of some programming language the modal operator $[\alpha]$, formulas of the form $[\alpha]\phi$ being read "every execution of the program α from the present state leads to a state bearing the formula ϕ". Completeness and decidability results for the standard version of PDL in which programs are built up from program variables and tests by means of the operations of composition, union and iteration are given in [15,16]. A number of interesting variants have been obtained by extending or restricting the syntax or the semantics of PDL in different ways [7,9,14,18].

Some of these variants extend the ordinary semantics of PDL by considering sets W of states structured by means of a function $\star$ from the set of all pairs of states into the set of all states [5,11–13]: the state x is the result of applying the function $\star$ to the states y, z iff the information concerning x can be separated in a first part concerning y and a second part concerning z. The binary function $\star$ considered in [5,11] has its origin in the addition of an extra binary operation of fork denoted ∇ in relation algebras: in [5, Sect. 2], whenever x and y are related via R and z and t are related via S, states in $x \star z$ and states in $y \star t$ are related via $R \nabla S$ whereas in [11, Chap. 1], whenever x and y are related via R and x and z are related via S, x and states in $y \star z$ are related via $R \nabla S$.

This addition of fork in relation algebras gives rise to a variant of PDL which includes the program operation of fork denoted Δ. In this variant, for all programs α and β, one can use the modal operator $[\alpha \Delta \beta]$, formulas of the form $[\alpha \Delta \beta]\phi$ being read "every execution in parallel of the programs α and β from the

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 17–34, 2018.
https://doi.org/10.1007/978-3-319-73579-5_2

present state leads to a state bearing the formula ϕ". The binary operation of fork ∇ considered in Benevides *et al.* [5, Sect. 2] gives rise to $PRSPDL$, a variant of PDL with fork whose axiomatization is still open. We devote this paper to the axiomatization and the computability of PDL_0^{Δ}, a variant of iteration-free PDL with fork whose semantics is based on the interpretation of the binary operation of fork ∇ considered in Frias [11, Chap. 1].

The difficulty in axiomatizing or deciding $PRSPDL$ or PDL_0^{Δ} originates in the fact that the program operations of fork considered above are not modally definable in the ordinary language of PDL. We overcome this difficulty by means of tools and techniques developed in [1,3,4]. Our results are based on the following: although fork is not modally definable, it becomes definable in a modal language strengthened by the introduction of propositional quantifiers. Instead of using axioms to define the program operation of fork in the language of PDL enlarged with propositional quantifiers, we add an unorthodox rule of proof that makes the canonical model standard for the program operation of fork and we use large programs for the proof of the Truth Lemma.

We will first present the syntax (Sect. 2) and the semantics (Sect. 3) of PDL_0^{Δ} and continue with results concerning the expressivity of PDL_0^{Δ} (Sect. 4), the axiomatization/completeness of PDL_0^{Δ} (Sects. 5 and 6) and the decidability of PDL_0^{Δ} (Sect. 7). We assume the reader is at home with tools and techniques in modal logic and dynamic logic. For more on this, see [6,15]. The proofs of our results can be found in [2].

2 Syntax

This section presents the syntax of PDL_0^{Δ}. As usual, we will follow the standard rules for omission of the parentheses.

Definition 1 (Programs and formulas). *The set PRG of all programs and the set FRM of all formulas are inductively defined as follows:*

- $\alpha, \beta ::= a \mid (\alpha;\beta) \mid (\alpha \Delta \beta) \mid \phi?$;
- $\phi, \psi ::= p \mid \bot \mid \neg\phi \mid (\phi \vee \psi) \mid [\alpha]\phi \mid (\phi \circ \psi) \mid (\phi \rhd \psi) \mid (\phi \lhd \psi)$;

where a ranges over a countably infinite set of program variables and p ranges over a countably infinite set of propositional variables.

We will use $\alpha, \beta, \ldots$ for programs and $\phi, \psi, \ldots$ for formulas. The Boolean constructs for formulas are defined as usual. A number of other constructs for formulas can be defined in terms of the primitive ones as follows.

Definition 2 (Abbreviations). *The modal constructs for formulas $\langle\cdot\rangle\cdot$, $(\cdot\bar{\circ}\cdot)$, $(\cdot\bar{\rhd}\cdot)$ and $(\cdot\bar{\lhd}\cdot)$ are defined as follows: $\langle\alpha\rangle\phi ::= \neg[\alpha]\neg\phi$; $(\phi\bar{\circ}\psi) ::= \neg(\neg\phi \circ \neg\psi)$; $(\phi\bar{\rhd}\psi) ::= \neg(\neg\phi \rhd \neg\psi)$; $(\phi\bar{\lhd}\psi) ::= \neg(\neg\phi \lhd \neg\psi)$. Moreover, for all formulas ϕ, let $\phi^0 ::= \neg\phi$ and $\phi^1 ::= \phi$.*

It is well worth noting that programs and formulas are finite strings of symbols coming from a countable alphabet. It follows that there are countably many programs and countably many formulas. The construct $\cdot;\cdot$ comes from the class of algebras of binary relations [19]: the program $\alpha;\beta$ firstly executes α and secondly executes β. As for the construct $\cdot\varDelta\cdot$, it comes from the class of proper fork algebras [11, Chap. 1]: the program $\alpha\varDelta\beta$ performs a kind of parallel execution of α and β. The construct $[\cdot]\cdot$ comes from the language of PDL [10,15]: the formula $[\alpha]\phi$ says that "every execution of α from the present state leads to a state bearing the information ϕ". As for the constructs $\cdot\circ\cdot$, $\cdot\triangleright\cdot$ and $\cdot\triangleleft\cdot$, they come from the language of conjugated arrow logic [8,17]: the formula $\phi\circ\psi$ says that "the present state is a combination of states bearing the information ϕ and ψ", the formula $\phi\triangleright\psi$ says that "the present state can be combined to its left with a state bearing the information ϕ giving us a state bearing the information ψ" and the formula $\phi\triangleleft\psi$ says that "the present state can be combined to its right with a state bearing the information ψ giving us a state bearing the information ϕ".

Example 1. The formula $[a\varDelta b](p\circ q)$ says that "the parallel execution of a and b from the present state always leads to a state resulting from the combination of states bearing the information p and q".

Obviously, programs are built up from program variables and tests by means of the constructs $\cdot;\cdot$ and $\cdot\varDelta\cdot$. Let $\alpha(\phi_1?,\ldots,\phi_n?)$ be a program with $(\phi_1?,\ldots,\phi_n?)$ a sequence of some of its tests. The result of the replacement of $\phi_1?,\ldots,\phi_n?$ in their places with other tests $\psi_1?,\ldots,\psi_n?$ is another program which will be denoted $\alpha(\psi_1?,\ldots,\psi_n?)$. Now, we introduce the function f from the set of all programs into itself defined as follows.

Definition 3 (Test insertion). *Let f be the function from the set of all programs into itself inductively defined as follows:*

- $f(a) = a$;
- $f(\alpha;\beta) = f(\alpha);\top?;f(\beta)$;
- $f(\alpha\varDelta\beta) = (f(\alpha);\top?)\varDelta(f(\beta);\top?)$;
- $f(\phi?) = \phi?$.

Example 2. If $\alpha = a\varDelta b$, $f(\alpha) = (a;\top?)\varDelta(b;\top?)$.

Now, we introduce parametrized actions and admissible forms.

Definition 4 (Parametrized actions and admissible forms). *The set PAR of all parametrized actions and the set ADM of all admissible forms are inductively defined as follows:*

- $\breve{\alpha},\breve{\beta} ::= (\breve{\alpha};\beta) \mid (\alpha;\breve{\beta}) \mid (\breve{\alpha}\varDelta\beta) \mid (\alpha\varDelta\breve{\beta}) \mid \neg\breve{\phi}?$;
- $\breve{\phi},\breve{\psi} ::= \sharp \mid [\breve{\alpha}]\bot \mid (\breve{\phi}\bar{\circ}\psi) \mid (\phi\bar{\circ}\breve{\psi}) \mid (\breve{\phi}\bar{\triangleright}\psi) \mid (\phi\bar{\triangleright}\breve{\psi}) \mid (\breve{\phi}\bar{\triangleleft}\psi) \mid (\phi\bar{\triangleleft}\breve{\psi})$;

where $\sharp$ is a new propositional variable, α,β range over PRG and ϕ,ψ range over FRM.

We will use $\breve{\alpha}, \breve{\beta}, \ldots$ for parametrized actions and $\breve{\phi}, \breve{\psi}, \ldots$ for admissible forms. It is well worth noting that parametrized actions and admissible forms are finite strings of symbols coming from a countable alphabet. It follows that there are countably many parametrized actions and countably many admissible forms. Remark that in each expression $e\breve{x}p$ (a parametrized action, or an admissible form), $\sharp$ has a unique occurrence. The result of the replacement of $\sharp$ in its place in $e\breve{x}p$ with a formula ϕ is an expression which will be denoted $e\breve{x}p(\phi)$.

Example 3. For all programs α, $\alpha; \neg[\neg\sharp?]\bot?$ is a parametrized action whereas for all formulas ϕ, $\phi\bar{\circ}[\neg\sharp?]\bot$ is an admissible form.

3 Semantics

Our task is now to present the semantics of PDL_0^{Δ}.

Definition 5 (Frames). *A frame is a 3-tuple $\mathcal{F} = (W, R, \star)$ 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 and $\star$ is a function from the set of all pairs of states into the set of all sets of states.*

We will use $x, y, \ldots$ for states. The set W of states in a frame $\mathcal{F} = (W, R, \star)$ is to be regarded as the set of all possible states in a computation process, the function R from the set of all program variables into the set of all binary relations between states associates with each program variable a the binary relation $R(a)$ on W with $xR(a)y$ meaning that "y can be reached from x by performing program variable a" and the function $\star$ from the set of all pairs of states into the set of all sets of states associates with each pair (x, y) of states the subset $x \star y$ of W with $z \in x \star y$ meaning that "z is a combination of x and y".

Definition 6 (Valuations and models). *A model on the frame $\mathcal{F} = (W, R, \star)$ is a 4-tuple $\mathcal{M} = (W, R, \star, V)$ 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 the model $\mathcal{M} = (W, R, \star, V)$, the valuation V associates with each propositional variable p the subset $V(p)$ of W with $x \in V(p)$ meaning that "propositional variable p is true at state x in $\mathcal{M}$". We now define the property "state y can be reached from state x by performing program α in $\mathcal{M}$"—in symbols $xR_{\mathcal{M}}(\alpha)y$—and the property "formula ϕ is true at state x in $\mathcal{M}$"—in symbols $x \in V_{\mathcal{M}}(\phi)$.

Definition 7 (Accessibility via programs and truth of formulas). *In model $\mathcal{M} = (W, R, \star, V)$, $R_{\mathcal{M}} : \alpha \mapsto R_{\mathcal{M}}(\alpha) \subseteq W \times W$ and $V_{\mathcal{M}} : \phi \mapsto V_{\mathcal{M}}(\phi) \subseteq W$ are inductively defined 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 $zR_{\mathcal{M}}(\beta)y$;*

- $xR_{\mathcal{M}}(\alpha \Delta \beta)y$ *iff there exists* $z, t \in W$ *such that* $xR_{\mathcal{M}}(\alpha)z$, $xR_{\mathcal{M}}(\beta)t$ *and* $y \in z \star t$;
- $xR_{\mathcal{M}}(\phi?)y$ *iff* $x = y$ *and* $y \in V_{\mathcal{M}}(\phi)$;
- $x \in V_{\mathcal{M}}(p)$ *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* $x \in V_{\mathcal{M}}(\phi)$, *or* $x \in V_{\mathcal{M}}(\psi)$;
- $x \in V_{\mathcal{M}}([\alpha]\phi)$ *iff for all* $y \in W$, *if* $xR_{\mathcal{M}}(\alpha)y$, $y \in V_{\mathcal{M}}(\phi)$;
- $x \in V_{\mathcal{M}}(\phi \circ \psi)$ *iff there exists* $y, z \in W$ *such that* $x \in y \star z$, $y \in V_{\mathcal{M}}(\phi)$ *and* $z \in V_{\mathcal{M}}(\psi)$;
- $x \in V_{\mathcal{M}}(\phi \rhd \psi)$ *iff there exists* $y, z \in W$ *such that* $z \in y \star x$, $y \in V_{\mathcal{M}}(\phi)$ *and* $z \in V_{\mathcal{M}}(\psi)$;
- $x \in V_{\mathcal{M}}(\phi \lhd \psi)$ *iff there exists* $y, z \in W$ *such that* $y \in x \star z$, $y \in V_{\mathcal{M}}(\phi)$ *and* $z \in V_{\mathcal{M}}(\psi)$.

It follows that

Proposition 1. *Let* $\mathcal{M} = (W, R, \star, V)$ *be a model. For all* $x \in W$, *we have:* $x \in V_{\mathcal{M}}(\langle\alpha\rangle\phi)$ *iff there exists* $y \in W$ *such that* $xR_{\mathcal{M}}(\alpha)y$ *and* $y \in V_{\mathcal{M}}(\phi)$; $x \in V_{\mathcal{M}}(\phi \bar{\circ} \psi)$ *iff for all* $y, z \in W$, *if* $x \in y \star z$, $y \in V_{\mathcal{M}}(\phi)$, *or* $z \in V_{\mathcal{M}}(\psi)$; $x \in V_{\mathcal{M}}(\phi \bar{\rhd} \psi)$ *iff for all* $y, z \in W$, *if* $z \in y \star x$, $y \in V_{\mathcal{M}}(\phi)$, *or* $z \in V_{\mathcal{M}}(\psi)$; $x \in V_{\mathcal{M}}(\phi \bar{\lhd} \psi)$ *iff for all* $y, z \in W$, *if* $y \in x \star z$, $y \in V_{\mathcal{M}}(\phi)$, *or* $z \in V_{\mathcal{M}}(\psi)$.

Example 4. Let $\mathcal{M} = (W, R, \star, V)$ be the model defined by:

- $W = \{x, y, z, t\}$;
- $R(a) = \{(x, y)\}$, $R(b) = \{(x, z)\}$, otherwise R is the empty function;
- $y \star z = \{t\}$, otherwise $\star$ is the empty function;
- $V(p) = \{y\}$, $V(q) = \{z\}$, otherwise V is the empty function.

Obviously, $xR_{\mathcal{M}}(a \Delta b)t$ and $t \in V_{\mathcal{M}}(p \circ q)$. Hence, $x \in V_{\mathcal{M}}(\langle a \Delta b\rangle(p \circ q))$.

We now define the property "state z can be reached from state x by performing parametrized action $\breve{\alpha}$ via state y in $\mathcal{M}$"—in symbols $xR_{\mathcal{M}}(\breve{\alpha}, y)z$—and the property "admissible form $\breve{\phi}$ is true at state x via state y in $\mathcal{M}$"—in symbols $x \in V_{\mathcal{M}}(\breve{\phi}, y)$.

Definition 8 (Accessibility via parametrized actions and truth of admissible forms). *In model* $\mathcal{M} = (W, R, \star, V)$, $R_{\mathcal{M}} : (\breve{\alpha}, y) \mapsto R_{\mathcal{M}}(\breve{\alpha}, y) \subseteq W \times W$ *and* $V_{\mathcal{M}} : (\breve{\phi}, y) \mapsto V_{\mathcal{M}}(\breve{\phi}, y) \subseteq W$ *are inductively defined as follows:*

- $xR_{\mathcal{M}}(\breve{\alpha}; \beta, y)z$ *iff there exists* $t \in W$ *such that* $xR_{\mathcal{M}}(\breve{\alpha}, y)t$ *and* $tR_{\mathcal{M}}(\beta)z$;
- $xR_{\mathcal{M}}(\alpha; \breve{\beta}, y)z$ *iff there exists* $t \in W$ *such that* $xR_{\mathcal{M}}(\alpha)t$ *and* $tR_{\mathcal{M}}(\breve{\beta}, y)z$;
- $xR_{\mathcal{M}}(\breve{\alpha} \Delta \beta, y)z$ *iff there exists* $t, u \in W$ *such that* $xR_{\mathcal{M}}(\breve{\alpha}, y)t$, $xR_{\mathcal{M}}(\beta)u$ *and* $z \in t \star u$;
- $xR_{\mathcal{M}}(\alpha \Delta \breve{\beta}, y)z$ *iff there exists* $t, u \in W$ *such that* $xR_{\mathcal{M}}(\alpha)t$, $xR_{\mathcal{M}}(\breve{\beta}, y)u$ *and* $z \in t \star u$;
- $xR_{\mathcal{M}}(\neg\breve{\phi}?, y)z$ *iff* $x = z$ *and* $z \in V_{\mathcal{M}}(\breve{\phi}, y)$;

- $x \in V_{\mathcal{M}}(\sharp, y)$ *iff* $x = y$;
- $x \in V_{\mathcal{M}}([\breve{\alpha}]\bot, y)$ *iff there exists* $z \in W$ *such that* $xR_{\mathcal{M}}(\breve{\alpha}, y)z$;
- $x \in V_{\mathcal{M}}(\breve{\phi}\bar{\circ}\psi, y)$ *iff there exists* $z, t \in W$ *such that* $x \in z \star t$, $z \in V_{\mathcal{M}}(\breve{\phi}, y)$ *and* $t \notin V_{\mathcal{M}}(\psi)$;
- $x \in V_{\mathcal{M}}(\phi\bar{\circ}\breve{\psi}, y)$ *iff there exists* $z, t \in W$ *such that* $x \in z \star t$, $z \notin V_{\mathcal{M}}(\phi)$ *and* $t \in V_{\mathcal{M}}(\breve{\psi}, y)$;
- $x \in V_{\mathcal{M}}(\breve{\phi}\bar{\triangleright}\psi, y)$ *iff there exists* $z, t \in W$ *such that* $t \in z \star x$, $z \in V_{\mathcal{M}}(\breve{\phi}, y)$ *and* $t \notin V_{\mathcal{M}}(\psi)$;
- $x \in V_{\mathcal{M}}(\phi\bar{\triangleright}\breve{\psi}, y)$ *iff there exists* $z, t \in W$ *such that* $t \in z \star x$, $z \notin V_{\mathcal{M}}(\phi)$ *and* $t \in V_{\mathcal{M}}(\breve{\psi}, y)$;
- $x \in V_{\mathcal{M}}(\breve{\phi}\bar{\triangleleft}\psi, y)$ *iff there exists* $z, t \in W$ *such that* $z \in x \star t$, $z \in V_{\mathcal{M}}(\breve{\phi}, y)$ *and* $t \notin V_{\mathcal{M}}(\psi)$;
- $x \in V_{\mathcal{M}}(\phi\bar{\triangleleft}\breve{\psi}, y)$ *iff there exists* $z, t \in W$ *such that* $z \in x \star t$, $z \notin V_{\mathcal{M}}(\phi)$ *and* $t \in V_{\mathcal{M}}(\breve{\psi}, y)$;

It follows that

Proposition 2. *Let* $\mathcal{M} = (W, R, \star, V)$ *be a model. Let* $\breve{\alpha}$ *be a parametrized action. For all* $x, z \in W$, *the following conditions are equivalent:* $xR_{\mathcal{M}}(\breve{\alpha}(\phi))z$; *there exists* $y \in W$ *such that* $xR_{\mathcal{M}}(\breve{\alpha}, y)z$ *and* $y \notin V_{\mathcal{M}}(\phi)$. *Let* $\breve{\phi}$ *be an admissible form. For all* $x \in W$, *the following conditions are equivalent:* $x \in V_{\mathcal{M}}(\breve{\phi}(\psi))$; *for all* $y \in W$, *if* $x \in V_{\mathcal{M}}(\breve{\phi}, y)$, $y \in V_{\mathcal{M}}(\psi)$.

The concept of validity is defined in the usual way as follows.

Definition 9 (Validity). *We shall say that a formula* ϕ *is valid in a model* $\mathcal{M}$, *in symbols* $\mathcal{M} \models \phi$, *iff* $V_{\mathcal{M}}(\phi) = W$. *A formula* ϕ *is said to be valid in a 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 valid in a class* $\mathcal{C}$ *of frames, in symbols* $\mathcal{C} \models \phi$, *iff for all frames* $\mathcal{F}$ *in* $\mathcal{C}$, $\mathcal{F} \models \phi$.

For technical reasons, we now consider three particular classes of frames.

Definition 10 (Separated, deterministic or serial frames). *A frame* $\mathcal{F} = (W, R, \star)$ *is said to be separated iff for all* $x, y, z, t, u \in W$, *if* $u \in x \star y$ *and* $u \in z \star t$, $x = z$ *and* $y = t$. *We shall say that a frame* $\mathcal{F} = (W, R, \star)$ *is deterministic iff for all* $x, y, z, t \in W$, *if* $z \in x \star y$ *and* $t \in x \star y$, $z = t$. *A frame* $\mathcal{F} = (W, R, \star)$ *is said to be serial iff for all* $x, y \in W$, *there exists* $z \in W$ *such that* $z \in x \star y$.

In separated frames, there is at most one way to decompose a given state; in deterministic frames, there is at most one way to combine two given states; in serial frames, it is always possible to combine two given states. Frias [11, Chap. 1] only considers separated, deterministic and serial frames. Here are some valid formulas and admissible rules of proof.

Proposition 3 (Validity). *The following formulas are valid in the class of all frames:*

$(A1)$ $[\alpha](\phi \to \psi) \to ([\alpha]\phi \to [\alpha]\psi)$;
$(A2)$ $\langle \alpha; \beta \rangle \phi \leftrightarrow \langle \alpha \rangle \langle \beta \rangle \phi$;
$(A3)$ $\langle \alpha \Delta \beta \rangle \phi \to \langle \alpha \rangle ((\phi \wedge \psi) \triangleleft \top) \vee \langle \beta \rangle (\top \triangleright (\phi \wedge \neg \psi))$;
$(A4)$ $\langle \phi ? \rangle \psi \leftrightarrow \phi \wedge \psi$;
$(A5)$ $(\phi \to \psi) \bar{\circ} \chi \to (\phi \bar{\circ} \chi \to \psi \bar{\circ} \chi)$;
$(A6)$ $\phi \bar{\circ} (\psi \to \chi) \to (\phi \bar{\circ} \psi \to \phi \bar{\circ} \chi)$;
$(A7)$ $(\phi \to \psi) \bar{\triangleright} \chi \to (\phi \bar{\triangleright} \chi \to \psi \bar{\triangleright} \chi)$;
$(A8)$ $\phi \bar{\triangleright} (\psi \to \chi) \to (\phi \bar{\triangleright} \psi \to \phi \bar{\triangleright} \chi)$;
$(A9)$ $(\phi \to \psi) \bar{\triangleleft} \chi \to (\phi \bar{\triangleleft} \chi \to \psi \bar{\triangleleft} \chi)$;
$(A10)$ $\phi \bar{\triangleleft} (\psi \to \chi) \to (\phi \bar{\triangleleft} \psi \to \phi \bar{\triangleleft} \chi)$;
$(A11)$ $\phi \circ \neg(\phi \triangleright \neg \psi) \to \psi$;
$(A12)$ $\phi \triangleright \neg(\phi \circ \neg \psi) \to \psi$;
$(A13)$ $\neg(\neg \phi \triangleleft \psi) \circ \psi \to \phi$;
$(A14)$ $\neg(\neg \phi \circ \psi) \triangleleft \psi \to \phi$;
$(A15)$ $[(\alpha; \phi ?) \Delta (\beta; \psi ?)](\phi \circ \psi)$;
$(A16)$ $\langle \alpha(\phi ?) \rangle \psi \to \langle \alpha((\phi \wedge \chi)?) \rangle \psi \vee \langle \alpha((\phi \wedge \neg \chi)?) \rangle \psi$;
$(A17)$ $\langle f(\alpha) \rangle \phi \leftrightarrow \langle \alpha \rangle \phi$.

Proposition 4 (Validity). *The following formula is valid in the class of all separated frames:*

$(A18)$ $p \circ q \to (p \bar{\circ} \bot) \wedge (\bot \bar{\circ} q)$.

Proposition 5 (Admissibility). *The following rules of proof preserve validity in the class of all frames:*

(MP) *from* ϕ *and* $\phi \to \psi$, *infer* ψ;
(N) *from* ϕ, *infer* $[\alpha]\phi$; *from* ϕ, *infer* $\phi \bar{\circ} \psi$; *from* ϕ, *infer* $\psi \bar{\circ} \phi$.

$(A1)$ is the distribution axiom of PDL, $(A2)$ is the composition axiom, $(A4)$ is the test axiom, $(A5)$–$(A10)$ are the distribution axioms of conjugated arrow logic and $(A11)$–$(A14)$ are the tense axioms of conjugated arrow logic whereas $(A3)$ and $(A15)$–$(A18)$ are axioms concerning specific properties of the program operation of fork or the constructs $\cdot \circ \cdot$, $\cdot \triangleright \cdot$ and $\cdot \triangleleft \cdot$. (MP) is the modus ponens rule of proof and (N) is the necessitation rule of proof. They are probably familiar to the reader. As for the following rule of proof, it concerns specific properties of the program operation of fork and the constructs $\cdot \triangleright \cdot$ and $\cdot \triangleleft \cdot$.

Proposition 6 (Admissibility). *The following rule of proof preserves validity in the class of all separated frames:*

(FOR) *from* $\{\breve{\phi}(\langle \alpha \rangle ((\psi \wedge p) \triangleleft \top) \vee \langle \beta \rangle (\top \triangleright (\psi \wedge \neg p)))$: p *is a propositional variable*$\}$, *infer* $\breve{\phi}(\langle \alpha \Delta \beta \rangle \psi)$.

There is an important point we should make: (FOR) is an infinitary rule of proof, i.e. it has an infinite set of formulas as preconditions. In some ways, it is similar to the rule for intersection from [3,4].

4 Expressivity

This section studies the expressivity of PDL_0^{Δ}.

Definition 11 (Modal definability). *Let $\mathcal{C}$ be a class of frames. We shall say that $\mathcal{C}$ is modally definable by the formula ϕ iff for all frames $\mathcal{F}$, $\mathcal{F}$ is in $\mathcal{C}$ iff $\mathcal{F} \models \phi$.*

The following propositions show elementary classes of frames that are modally definable.

Proposition 7. *The elementary classes of frames defined by the first-order sentences in the hereunder table are modally definable by the associated formulas.*

1.	$\forall x\ \exists y\ y \in x \star x$	$\langle \top ? \Delta \top ? \rangle \top$
2.	$\forall x\ \forall y\ \forall z\ (y \in x \star x \land z \in x \star x \to y = z)$	$\langle \top ? \Delta \top ? \rangle p \to [\top ? \Delta \top ?] p$
3.	$\forall x\ \forall y\ (y \in x \star x \to x \in x \star y)$	$p \to [\top ? \Delta \top ?](p \rhd p)$
4.	$\forall x\ \forall y\ (y \in x \star x \to x \in y \star x)$	$p \to [\top ? \Delta \top ?](p \lhd p)$
5.	$\forall x\ \forall y\ \forall z\ (z \in x \star y \leftrightarrow z \in y \star x)$	$p \circ q \leftrightarrow q \circ p$
6.	$\forall x\ \exists y\ \exists z\ x \in y \star z$	$\top \circ \top$
7.	$\forall x\ \exists y\ \exists z\ y \in z \star x$	$\top \rhd \top$
8.	$\forall x\ \exists y\ \exists z\ z \in x \star y$	$\top \lhd \top$
9.	$\forall x\ \forall y\ \forall z\ \forall t\ (t \in (x \star y) \star z \leftrightarrow t \in x \star (y \star z))$	$(p \circ q) \circ r \leftrightarrow p \circ (q \circ r)$
10.	$\forall x\ \forall y\ \forall z\ x \notin y \star z$	$\bot \bar{\circ} \bot$

Proposition 8. *The class of all separated frames is modally definable by the formula $p \circ q \to (p \bar{\circ} \bot) \land (\bot \bar{\circ} q)$.*

The following proposition shows an elementary class of frames that is not modally definable.

Proposition 9. *The class of all deterministic frames is not modally definable.*

As for the class of all serial frames, its modal definability is still open. In other respect, the formula $\langle \phi ? \rangle \psi \leftrightarrow \phi \land \psi$, being valid in the class of all frames, seems to indicate that for all formulas, there exists an equivalent test-free formula. It is interesting to observe that this assertion is false.

Proposition 10. *For all test-free formulas ϕ, $\langle \top ? \Delta \top ? \rangle \top \leftrightarrow \phi$ is not valid in the class of all separated deterministic frames.*

The following proposition illustrates the fact that the program operation of fork cannot be defined from the fork-free fragment of the language.

Proposition 11. *Let a be a program variable. For all fork-free formulas ϕ, $\langle a \Delta a \rangle \top \leftrightarrow \phi$ is not valid in the class of all separated deterministic frames.*

The following proposition illustrates the fact that, in the presence of propositional quantifiers, the program operation of fork becomes definable from the fork-free fragment of the language in the class of all separated frames.

Proposition 12. *Let $\mathcal{M} = (W, R, \star, V)$ be a separated model and $x \in W$. For all admissible forms $\breve{\phi}$, for all programs α, β, for all formulas ψ and for all propositional variables p, if p does not occur in $\breve{\phi}, \alpha, \beta, \psi$, the following conditions are equivalent:* (1) *$x \in V_{\mathcal{M}}(\breve{\phi}(\langle \alpha \Delta \beta \rangle \psi))$;* (2) *for all $V' : q \mapsto V'(q) \subseteq W$, if $V' \sim_p V$, $x \in V_{(W,R,\star,V')}(\breve{\phi}(\langle \alpha \rangle ((\psi \wedge p) \lhd \top) \vee \langle \beta \rangle (\top \rhd (\psi \wedge \neg p))))$.*

More precisely, in the presence of propositional quantifiers, the formulas $\langle \alpha \Delta \beta \rangle \phi$ and $\forall p (\langle \alpha \rangle ((\phi \wedge p) \lhd \top) \vee \langle \beta \rangle (\top \rhd (\phi \wedge \neg p)))$ are logically equivalent in the class of all separated frames. The implication $\langle \alpha \Delta \beta \rangle \phi \to \forall p (\langle \alpha \rangle ((\phi \wedge p) \lhd \top) \vee \langle \beta \rangle (\top \rhd (\phi \wedge \neg p)))$ can be expressed without propositional quantifiers by formulas: $\langle \alpha \Delta \beta \rangle \phi \to \langle \alpha \rangle ((\phi \wedge \psi) \lhd \top) \vee \langle \beta \rangle (\top \rhd (\phi \wedge \neg \psi)))$. See axiom $(A3)$ in Proposition 3. As for the implication $\forall p (\langle \alpha \rangle ((\phi \wedge p) \lhd \top) \vee \langle \beta \rangle (\top \rhd (\phi \wedge \neg p))) \to \langle \alpha \Delta \beta \rangle \phi$, it can be expressed by a rule of proof. The simplest form of such a rule of proof is: from $\{\langle \alpha \rangle ((\phi \wedge p) \lhd \top) \vee \langle \beta \rangle (\top \rhd (\phi \wedge \neg p)) : p$ is a propositional variable$\}$, infer $\langle \alpha \Delta \beta \rangle \phi$. See Proposition 6.

5 Axiom System

We now define PDL_0^{Δ}.

Definition 12 (PDL_0^{Δ}). *Let PDL_0^{Δ} be the least set of formulas that contains all instances of propositional tautologies, that contains the formulas* $(A1)$–$(A18)$ *considered in Propositions 3 and 4 and that is closed under the rules of proof* (MP), (N) *and* (FOR) *considered in Propositions 5 and 6.*

It is easy to establish the soundness for PDL_0^{Δ}:

Proposition 13 (Soundness for PDL_0^{Δ}). *Let ϕ be a formula. If $\phi \in PDL_0^{\Delta}$, ϕ is valid in the class of all separated frames.*

The completeness for PDL_0^{Δ} is more difficult to establish and we defer proving it till next section. In the meantime, it is well worth noting that for all separated models $\mathcal{M} = (W, R, \star, V)$ and for all $x \in W$, $\{\phi : x \in V_{\mathcal{M}}(\phi)\}$ is a set of formulas that contains PDL_0^{Δ} and that is closed under the rule of proof (MP). Now, we introduce theories.

Definition 13 (Theories). *A set S of formulas is said to be a theory iff $PDL_0^{\Delta} \subseteq S$ and S is closed under the rules of proof* (MP) *and* (FOR).

We will use $S, T, \ldots$ for theories. Obviously, the least theory is PDL_0^{Δ} and the greatest theory is the set of all formulas. Not surprisingly, we have

Lemma 1. *Let S be a theory. The following conditions are equivalent: S is equal to the set of all formulas; there exists a formula ϕ such that $\phi \in S$ and $\neg\phi \in S$; $\bot \in S$.*

Referring to Lemma 1, we define what it means for a theory to be consistent.

Definition 14 (Consistency of theories). *We shall say that a theory S is consistent iff for all formulas ϕ, $\phi \notin S$, or $\neg\phi \notin S$.*

By Lemma 1, there is only one inconsistent theory: the set of all formulas. Now, we define what it means for a theory to be maximal.

Definition 15 (Maximality of theories). *A theory S is said to be maximal iff for all formulas ϕ, $\phi \in S$, or $\neg\phi \in S$.*

We will use the following lemma without explicit reference:

Lemma 2. *Let S be a maximal consistent theory. We have: $\bot \notin S$; for all formulas ϕ, $\neg\phi \in S$ iff $\phi \notin S$; for all formulas ϕ, ψ, $\phi \vee \psi \in S$ iff $\phi \in S$, or $\psi \in S$.*

To know more about theories, we need yet another definition.

Definition 16 (Operations on theories). *If α is a program, ϕ is a formula and S is a theory, let $[\alpha]S = \{\phi : [\alpha]\phi \in S\}$ and $S + \phi = \{\psi : \phi \to \psi \in S\}$.*

In the next lemmas, we summarize some properties of theories.

Lemma 3. *Let S be a theory. For all programs α and for all formulas ϕ, we have:* (1) $[\phi?]S = S + \phi$; (2) $[\alpha]S$ *is a theory;* (3) $S + \phi$ *is a theory;* (4) ϕ, $S + \phi$ *is the least theory containing S and ϕ;* (5) $S + \phi$ *is consistent iff $\neg\phi \notin S$.*

Lemma 4. *Let S be a theory. If S is consistent, for all formulas ϕ, $S + \phi$ is consistent, or there exists a formula ψ such that the following conditions are satisfied: $S + \psi$ is consistent; $\psi \to \neg\phi \in PDL_0^{\Delta}$; if ϕ is in the form $\breve{\chi}(\langle\alpha\Delta\beta\rangle\theta)$ of a conclusion of the rule of proof (FOR), there exists a propositional variable p such that $\psi \to \neg\breve{\chi}(\langle\alpha\rangle((\theta \wedge p) \triangleleft \top) \vee \langle\beta\rangle(\top \triangleright (\theta \wedge \neg p))) \in PDL_0^{\Delta}$.*

Now, we are ready for the Lindenbaum Lemma.

Lemma 5 (Lindenbaum Lemma). *Let S be a theory. If S is consistent, there exists a maximal consistent theory containing S.*

To define the canonical frame of PDL_0^{Δ} in next section, we need yet another definition.

Definition 17 (Composition of theories). *If S and T are theory, let $S \circ T = \{\phi \circ \psi : \phi \in S \text{ and } \psi \in T\}$.*

To end this section, we present useful results.

Lemma 6. *Let ϕ, ψ be formulas and $\otimes \in \{\circ, \rhd, \rhd\}$. For all maximal consistent theories S, if $\phi \otimes \psi \in S$, for all formulas χ, we have:* (1) *$(\phi \wedge \chi) \otimes \psi \in S$, or there exists a formula θ such that the following conditions are satisfied: $(\phi \wedge \theta) \otimes \psi \in S$; $\theta \to \neg\chi \in PDL_0^{\Delta}$; if χ is in the form $\breve{\tau}(\langle\alpha\Delta\beta\rangle\mu)$ of a conclusion of the rule of proof (FOR), there exists a propositional variable p such that $\theta \to \neg\breve{\tau}(\langle\alpha\rangle((\mu \wedge p) \lhd \top) \vee \langle\beta\rangle(\top \rhd (\mu \wedge \neg p))) \in PDL_0^{\Delta}$;* (2) *$\phi \otimes (\psi \wedge \chi) \in S$, or there exists a formula θ such that the following conditions are satisfied: $\phi \otimes (\psi \wedge \theta) \in S$; $\theta \to \neg\chi \in PDL_0^{\Delta}$; if χ is in the form $\breve{\tau}(\langle\alpha\Delta\beta\rangle\mu)$ of a conclusion of the rule of proof (FOR), there exists a propositional variable p such that $\theta \to \neg\breve{\tau}(\langle\alpha\rangle((\mu \wedge p) \lhd \top) \vee \langle\beta\rangle(\top \rhd (\mu \wedge \neg p))) \in PDL_0^{\Delta}$.*

Lemma 7. *Let ϕ, ψ be formulas. For all maximal consistent theories S, we have:* (1) *if $\phi \circ \psi \in S$, there exists maximal consistent theories T, U such that $T \circ U \subseteq S$, $\phi \in T$ and $\psi \in U$;* (2) *if $\phi \rhd \psi \in S$, there exists maximal consistent theories T, U such that $T \circ S \subseteq U$, $\phi \in T$ and $\psi \in U$;* (3) *if $\phi \lhd \psi \in S$, there exists maximal consistent theories T, U such that $S \circ U \subseteq T$, $\phi \in T$ and $\psi \in U$.*

6 Completeness

Now, for the canonical frame of PDL_0^{Δ}.

Definition 18 (Canonical frame). *The canonical frame of PDL_0^{Δ} is the* 3*-tuple $\mathcal{F}_c = (W_c, R_c, \star_c)$ where W_c is the set of all maximal consistent theories, R_c is the function from the set of all program variables into the set of all binary relations between maximal consistent theories defined by $SR_c(a)T$ iff $[a]S \subseteq T$ and $\star_c$ is the function from the set of all pairs of maximal consistent theories into the set of all sets of maximal consistent theories defined by $U \in S \star_c T$ iff $S \circ T \subseteq U$.*

We show first that

Lemma 8. *$\mathcal{F}_c$ is separated.*

Now, for the canonical valuation of PDL_0^{Δ} and the canonical model of PDL_0^{Δ}.

Definition 19 (Canonical valuation and canonical model). *The canonical model of PDL_0^{Δ} is the* 4*-tuple $\mathcal{M}_c = (W_c, R_c, \star_c, V_c)$ where V_c is the canonical valuation of PDL_0^{Δ}, i.e. the function from the set of all propositional variables into the set of all sets of maximal consistent theories defined by $S \in V_c(p)$ iff $p \in S$.*

For the proof of the Truth Lemma, we have to consider large programs.

Definition 20 (Large programs). *The set of all large programs is inductively defined as follows:*

- $A ::= a \mid (A; B) \mid (A\Delta B) \mid \bar{S}?$;

where for all consistent theories S, $\bar{S}$ is a new symbol.

We will use $A, B, \dots$ for large programs. Let us be clear that each large program is a finite string of symbols coming from an uncountable alphabet. It follows that there are uncountably many large programs. For convenience, we omit the parentheses in accordance with the standard rules. It is essential that large programs are built up from program variables and symbols for consistent theories by means of the operations ; and Δ. Let $A(\bar{S_1}?, \dots, \bar{S_n}?)$ be a large program with $(\bar{S_1}, \dots, \bar{S_n})$ a sequence of some of its symbols for consistent theories. The result of the replacement of $\bar{S_1}, \dots, \bar{S_n}$ in their places with other symbols $\bar{T_1}, \dots, \bar{T_n}$ for consistent theories is another large program which will be denoted $A(\bar{T_1}?, \dots, \bar{T_n}?)$.

Definition 21 (Maximality of large programs). *A large program $A(\bar{S_1}?, \dots, \bar{S_n}?)$ with $(\bar{S_1}, \dots, \bar{S_n})$ the sequence of all its symbols for consistent theories will be defined to be maximal if the theories $S_1, \dots, S_n$ are maximal.*

It appears that large programs, maximal, or not, can be associated with a set of programs.

Definition 22 (Kernel function). *The kernel function* $\ker : A \mapsto \ker(A) \subseteq PRG$ *is inductively defined as follows:*

- $\ker(a) = \{a\}$;
- $\ker(A; B) = \{\alpha; \beta : \alpha \in \ker(A)$ *and* $\beta \in \ker(B)\}$;
- $\ker(A \Delta B) = \{\alpha \Delta \beta : \alpha \in \ker(A)$ *and* $\beta \in \ker(B)\}$;
- $\ker(\bar{S}) = \{\phi? : \phi \in S\}$.

The following lemmas play an important role in the proof of the completeness for PDL_0^Δ.

Lemma 9. *Let $\alpha(\phi?)$ be a program. For all maximal consistent theories S, if $\langle \alpha(\phi?) \rangle \top \in S$, for all formulas ψ, we have: $\langle \alpha((\phi \wedge \psi)?) \rangle \top \in S$, or there exists a formula χ such that the following conditions are satisfied: $\langle \alpha((\phi \wedge \chi)?) \rangle \top \in S$; $\chi \rightarrow \neg \psi \in PDL_0^\Delta$; if ψ is in the form $\breve{\theta}(\langle \beta \Delta \gamma \rangle \tau)$ of a conclusion of the rule of proof (FOR), there exists a propositional variable p such that $\chi \rightarrow \neg \breve{\theta}(\langle \beta \rangle ((\tau \wedge p) \lhd \top) \vee \langle \gamma \rangle (\top \rhd (\tau \wedge \neg p))) \in PDL_0^\Delta$.*

Lemma 10 (Diamond Lemma). *Let α be a program and ϕ be a formula. For all maximal consistent theories S, if $[\alpha]\phi \notin S$, there exists a maximal program A and there exists a maximal consistent theory T such that $f(\alpha) \in \ker(A)$, for all programs β, if $\beta \in \ker(A)$, $[\beta]S \subseteq T$ and $\phi \notin T$.*

With this established, we are ready for the Truth Lemma.

Lemma 11 (Truth Lemma). *Let α be a program. For all maximal consistent theories S, T, the following conditions are equivalent: $SR_{\mathcal{M}_c}(\alpha)T$; there exists a maximal program A such that $f(\alpha) \in \ker(A)$ and for all programs β, if $\beta \in \ker(A)$, $[\beta]S \subseteq T$. Let ϕ be a formula. For all maximal consistent theories S, the following conditions are equivalent: $S \in V_{\mathcal{M}_c}(\phi)$; $\phi \in S$.*

Now, we are ready for the completeness for PDL_0^Δ.

Proposition 14 (Completeness for PDL_0^Δ). *Let ϕ be a formula. If ϕ is valid in the class of all separated frames, $\phi \in PDL_0^\Delta$.*

7 Decidability

In this section, we prove that the logic completely axiomatized in the previous sections is decidable. We use the notation $\sim\phi$ which is defined by: $\sim\phi =$ if there exists a formula ψ such that $\phi = \neg\psi$ then ψ else $\neg\phi$. We use ν to denote an expression which may be either a program or a formula and $|\nu|$ to denote the number of occurrences of symbols in ν. The following size function provides a more semantical measure on programs.

Definition 23 (Size of programs). *Let* size *be the function from the set of all programs to* $\mathbb{N}$ *inductively defined as follows:*

- $\mathrm{size}(\phi?) = 0$;
- $\mathrm{size}(a) = 1$;
- $\mathrm{size}(\alpha;\beta) = \mathrm{size}(\alpha) + \mathrm{size}(\beta)$;
- $\mathrm{size}(\alpha\Delta\beta) = \min(\mathrm{size}(\alpha), \mathrm{size}(\beta)) + 1$.

Obviously, if $x\ R_{\mathcal{M}}(\alpha)\ y$ and $\mathrm{size}(\alpha) = 0$ then $x = y$. Now we decompose expressions into subexpressions, associating a *depth* to each subformula.

Definition 24 (Localized expression and decomposition). *A localized expression is a tuple $d\colon \nu$ where ν is an expression and $d \in \mathbb{N}$ is called the depth. Given any localized expression $d\colon \nu$, the decomposition $\mathrm{Cl}(d\colon \nu)$ of $d\colon \nu$ is the least set of localized expressions containing $d\colon \nu$ and closed by the application of the rules from Fig. 1. We write $\mathrm{Cl}(\phi)$ for $\mathrm{Cl}(0\colon \phi)$.*

$$\frac{d\colon \phi}{d\colon \sim\phi} \qquad \frac{d\colon \langle\alpha\rangle\phi}{d\colon \alpha \quad d+\mathrm{size}(\alpha)\colon \phi} \qquad \frac{d\colon \phi\vee\psi}{d\colon \phi \quad d\colon \psi}$$

$$\frac{d\colon \phi?}{d\colon \phi} \qquad \frac{d\colon \alpha;\beta}{d\colon \alpha \quad d+\mathrm{size}(\alpha)\colon \beta} \qquad \frac{d\colon \alpha\Delta\beta}{d\colon \alpha \quad d\colon \beta}$$

$$\frac{d\colon \phi\circ\psi}{d+1\colon \phi \quad d+1\colon \psi} \qquad \frac{d\colon \phi\rhd\psi}{d+1\colon \phi \quad d+1\colon \psi} \qquad \frac{d\colon \phi\lhd\psi}{d+1\colon \phi \quad d+1\colon \psi}$$

Fig. 1. Rules for the decomposition of localized programs and formulas

Lemma 12. *The cardinality of* $\mathrm{Cl}(\phi)$ *is linear in* $|\phi|$.

Lemma 13. $\max\{d \mid \exists\nu, d\colon \phi \in \mathrm{Cl}(\phi)\}$ *is linear in* $|\phi|$.

We now prove a strong finite model property for PDL_0^{Δ} interpreted over the class of all separated frames. The procedure SELECTION on the following page creates a model $\mathcal{M}_s$ from a model $\mathcal{M}_o$ satisfying a formula ϕ_0 at w_0. It uses the recursive procedure LINK described in Procedure 2.

Input: A formula ϕ_0, a model $\mathcal{M}_o = (W_o, R_o, \star_o, V_o)$ and an initial state $w_0 \in W_o$ such that $w_0 \in V_{\mathcal{M}_o}(\phi_0)$.
Result: A finite model $\mathcal{M}_s = (W_s, R_s, \star_s, V_s)$.
Data: A subset $K \subseteq W_s$ of marked nodes and an integer $n \in \mathbb{N}$.

initialisation
$n = 0$;
$W_s = \{(0, 0, w_0)\}$;
$R_s(a) = \emptyset$ for all $a \in \Pi_0$;
$(O, 0, w_0) \star_s (O, 0, w_0) = \emptyset$;
$K = \emptyset$;
end
while $K \neq W_s$ **do**
choose an unmarked state $(k, d, w) \in W_s \setminus K$;
while $(k, d, w) \notin K$ **do**
let $V_s(p) = \{(k_x, d_x, x) \in W_s \mid x \in V_o(p)\}$ for all $p \in \Phi_0$;
if *there exists* d': $\langle \alpha \rangle \phi \in \mathrm{Cl}(\phi_0)$ *such that* $\mathrm{size}(\alpha) > 0$, $d' \geq d$, $w \in V_{\mathcal{M}_o}(\langle \alpha \rangle \phi)$ *and* $(k, d, w) \notin V_{\mathcal{M}_s}(\langle \alpha \rangle \phi)$ **then**
choose y **s.t.** $w \, R_{\mathcal{M}_o}(\alpha) \, y$ and $y \in V_{\mathcal{M}_o}(\phi)$;
let $d_y = d + \mathrm{size}(\alpha)$;
let $n = n + 1$;
add (n, d_y, y) **to** W_s ;
call LINK $(\mathcal{M}_o, \mathcal{M}_s, n, (k, d, w), (n, d_y, y), \alpha)$;
else if *there exists* d': $\phi \circ \psi \in \mathrm{Cl}(\phi_0)$ *such that* $d' \geq d$, $w \in V_{\mathcal{M}_o}(\phi \circ \psi)$ *and there is no* $(k_x, d_x, x), (k_y, d_y, y) \in W_s$ *such that* $(k, d, w) \in (k_x, d_x, x) \star_s (k_y, d_y, y)$ **then**
choose x *and* y **s.t.** $w \in x \star_s y$, $x \in V_{\mathcal{M}_o}(\phi)$ and $y \in V_{\mathcal{M}_o}(\psi)$;
add $(n+1, d+1, x)$ and $(n+2, d+1, y)$ **to** W_s ;
add (k, d, w) **to** $(n+1, d+1, x) \star_s (n+2, d+1, y)$;
let $n = n + 2$;
else if *there exists* d': $\phi \rhd \psi \in \mathrm{Cl}(\phi_0)$ *such that* $d' \geq d$, $w \in V_{\mathcal{M}_o}(\phi \rhd \psi)$ *and* $(k, d, w) \notin V_{\mathcal{M}_s}(\phi \rhd \psi)$ **then**
choose x *and* y **s.t.** $y \in x \star_s w$, $x \in V_{\mathcal{M}_o}(\phi)$ and $y \in V_{\mathcal{M}_o}(\psi)$;
add $(n+1, d+1, x)$ and $(n+2, d+1, y)$ **to** W_s ;
add $(n+2, d+1, y)$ **to** $(n+1, d+1, x) \star_s (k, d, w)$;
let $n = n + 2$;
else if *there exists* d': $\phi \lhd \psi \in \mathrm{Cl}(\phi_0)$ *such that* $d' \geq d$, $w \in V_{\mathcal{M}_o}(\phi \lhd \psi)$ *and* $\mathcal{M}_s, (d, w) \lhd \notin V_\phi(\psi)$ **then**
choose x *and* y **s.t.** $x \in w \star_s y$, $x \in V_{\mathcal{M}_o}(\phi)$ and $y \in V_{\mathcal{M}_o}(\psi)$;
add $(n+1, d+1, x)$ and $(n+2, d+1, y)$ **to** W_s ;
add $(n+1, d+1, x)$ **to** $(k, d, w) \star_s (n+2, d+1, y)$;
let $n = n + 2$;
else
add (k, d, w) **to** K ;
end
end
end

Procedure 1. SELECTION

Input: Two models $\mathcal{M}_o = (W_o, R_o, \star_o, V_o)$ and $\mathcal{M}_s = (W_s, R_s, \star_s, V_s)$, an integer n, two states $(k_x, d_x, x), (k_y, d_y, y) \in W_s$ and a program α such that $x\ R_{\mathcal{M}_o}(\alpha)\ y$.

Result: $\mathcal{M}_s$ and n modified.

if *α is of the form $a \in \Pi_0$* **then**
  add $((k_x, d_x, x), (k_y, d_y, y))$ **to** $R_s(a)$;
else if *α is of the form $(\beta;\gamma)$* **then**
  if $\mathrm{size}(\beta) = 0$ **then**
    call LINK $(\mathcal{M}_o, \mathcal{M}_s, n, (k_x, d_x, x), (k_y, d_y, y), \gamma)$;
  else if $\mathrm{size}(\gamma) = 0$ **then**
    call LINK $(\mathcal{M}_o, \mathcal{M}_s, n, (k_x, d_x, x), (k_y, d_y, y), \beta)$;
  else
    choose z **s.t.** $x\ R_{\mathcal{M}_o}(\beta)\ z$ and $z\ R_{\mathcal{M}_o}(\gamma)\ y$;
    let $n = n + 1$;
    let $d_z = d_x + \mathrm{size}(\alpha)$;
    add (n, d_z, z) **to** W_s ;
    call LINK $(\mathcal{M}_o, \mathcal{M}_s, n, (k_x, d_x, x), (n, d_z, z), \beta)$;
    call LINK $(\mathcal{M}_o, \mathcal{M}_s, n, (n, d_z, z), (k_y, d_y, y), \gamma)$;
  end
else if *α is of the form $(\beta\Delta\gamma)$* **then**
  if $\mathrm{size}(\beta) = 0$ *and* $\mathrm{size}(\gamma) = 0$ **then**
    add (k_y, d_y, y) **to** $(k_x, d_x, x) \star_s (k_x, d_x, x)$;
  else if $\mathrm{size}(\beta) = 0$ **then**
    choose z **s.t.** $x\ R_{\mathcal{M}_o}(\gamma)\ z$ and $y \in x \star_o z$;
    let $n = n + 1$;
    let $d_z = \min(d_y + 1, d_x + \mathrm{size}(\gamma))$;
    add (n, d_z, z) **to** W_s ;
    add (k_y, d_y, y) **to** $(k_x, d_x, x) \star_s (n, d_z, z)$;
    call LINK $(\mathcal{M}_o, \mathcal{M}_s, n, (k_x, d_x, x), (n, d_z, z), \gamma)$;
  else if $\mathrm{size}(\gamma) = 0$ **then**
    choose w **s.t.** $x\ R_{\mathcal{M}_o}(\beta)\ w$ and $y \in w \star_o x$;
    let $n = n + 1$;
    let $d_w = \min(d_y + 1, d_x + \mathrm{size}(\beta))$;
    add (n, d_w, w) **to** W_s ;
    add (k_y, d_y, y) **to** $(n, d_w, w) \star_s (k_x, d_x, x)$;
    ;
    call LINK $(\mathcal{M}_o, \mathcal{M}_s, n, (k_x, d_x, x), (n, d_w, w), \beta)$;
  else
    choose w *and* z **s.t.** $x\ R_{\mathcal{M}_o}(\beta)\ w$, $x\ R_{\mathcal{M}_o}(\gamma)\ z$ and $y \in w \star_o z$;
    let $n = n + 2$;
    let $d_w = \min(d_y + 1, d_x + \mathrm{size}(\beta), d_x + \mathrm{size}(\gamma) + 1)$;
    let $d_z = \min(d_y + 1, d_x + \mathrm{size}(\gamma), d_x + \mathrm{size}(\beta) + 1)$;
    add $(n - 1, d_w, w)$ and (n, d_z, z) **to** W_s ;
    add (k_y, d_y, y) **to** $(n - 1, d_w, w) \star_s (n, d_z, z)$;
    call LINK $(\mathcal{M}_o, \mathcal{M}_s, n, (k_x, d_x, x), (n - 1, d_w, w), \beta)$;
    call LINK $(\mathcal{M}_o, \mathcal{M}_s, n, (k_x, d_x, x), (n, d_z, z), \gamma)$;
  end
end

Procedure 2. LINK

Lemma 14. *The procedure* SELECTION *terminates and the cardinality of* W_s *is exponential in* $|\phi_0|$.

Lemma 15. *Whenever* LINK *is called,* $d_y \leq d_x + \mathrm{size}(\alpha)$.

Lemma 16. *For all* $(k_y, d_y, y), (k_w, d_w, w), (k_z, d_z, z) \in W_s$, *such that* $(k_y, d_y, y) \in (k_w, d_w, w) \star_s (k_z, d_z, z)$ *then* $y \in w \star_o z$, $|d_y - d_w| \leq 1$, $|d_y - d_z| \leq 1$ *and* $|d_w - d_z| \leq 1$.

Lemma 17. *For all* $(k_x, d_x, x), (k_y, d_y, y) \in W_s$ *and all* α, *if* $(k_x, d_x, x)\ R_{\mathcal{M}_s}(\alpha)\ (k_y, d_y, y)$, *then* $d_y \leq d_x + \mathrm{size}(\alpha)$.

Lemma 18. *If* $\mathcal{M}_o$ *is separated, then* $\mathcal{M}_s$ *is separated too.*

Lemma 19 (Truth lemma). *If* $\mathcal{M}_o$ *is separated, then* $(0, 0, w_0) \in V_{\mathcal{M}_s}(\phi_0)$.

Proposition 15. *Any* PDL_0^{Δ} *formula* ϕ *satisfiable in a separated model is satisfiable in a separated finite model with a number of states bounded by an exponential in* $|\phi|$.

Since the model-checking problem for PDL_0^{Δ} is obviously polynomial in the size of the model, therefore we have the following corollary:

Corollary 1. *The satisfiability problem for* PDL_0^{Δ} *in the class of separated frames is decidable in non-deterministic exponential time.*

8 Conclusion

In modal logic, standard proofs of completeness for a given logic are usually based on the canonical frame construction consisting of the set of all maximal consistent sets of the logic equipped with standard definitions for the canonical accessibility relations. Since the program operation of fork considered in [11, Chap. 1] is not modally definable in the ordinary language of PDL, this method cannot work in our case. As a result, we have given an axiomatization of PDL_0^{Δ}, our variant of iteration-free PDL with fork, using an unorthodox rule of proof and we have proved its completeness using large programs. So, we have extended the canonical frame construction introducing new tools and techniques connected with an unorthodox rule of proof and large programs.

We anticipate a number of further investigations. First, there is the following general question: is it possible to eliminate the rule of proof (FOR) and to replace it with a finite set of additional axiom schemes? Second, more details on decidability/complexity issues would be relevant. Third, there is the question of the complete axiomatization of validity with respect to other classes of frames like the class of frames considered in [11, Chap. 1], i.e. the class of all separated, deterministic and serial frames. Fourth, is the validity problem with respect to the class of all separated, deterministic and serial frames decidable? If it is, what is its complexity? Fifth, it remains to see whether our approach can be extended to the full language of PDL with fork, this time with iteration.

A novelty in the paper is the proof that fork is modally definable in a language with propositional quantifiers and that the rule (FOR) in a sense simulates the quantifier rule for universal quantification in the context of the definition of fork. This is a new look on the nature of some context dependent rules of proof like (FOR). In some ways, (FOR) is similar to the rule for intersection from [3,4]. See also [1] for ideas about its elimination from the axiomatization of PDL_0^{Δ} we have given. We expect that our variant of the canonical frame construction can be applied to other logics, for instance $PRSPDL$, the variant of PDL with fork given rise by the binary operation of fork ∇ considered in Benevides *et al.* [5, Sect. 2] and whose axiomatization is still open.

Acknowledgement. Our research has been supported by the ANR project no. ANR-11-BS02-011.

References

1. Balbiani, P.: Eliminating unorthodox derivation rules in an axiom system for iteration-free PDL with intersection. Fundam. Inf. **56**, 211–242 (2003)
2. Balbiani, P., Boudou, J.: About the axiomatization and the computability of a variant of iteration-free PDL with fork. Institut de recherche en information de Toulouse (2017)
3. Balbiani, P., Vakarelov, D.: Iteration-free PDL with intersection: a complete axiomatization. Fundam. Inf. **45**, 173–194 (2001)
4. Balbiani, P., Vakarelov, D.: PDL with intersection of programs: a complete axiomatization. J. Appl. Non-Classical Logics **13**, 231–276 (2003)
5. Benevides, M., de Freitas, R., Viana, P.: Propositional dynamic logic with storing, recovering and parallel composition. Electron. Notes Theor. Comput. Sci. **269**, 95–107 (2011)
6. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press, Cambridge (2001)
7. Danecki, R.: Nondeterministic propositional dynamic logic with intersection is decidable. In: Skowron, A. (ed.) SCT 1984. LNCS, vol. 208, pp. 34–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-16066-3_5
8. Došen, K.: A brief survey of frames for the Lambek calculus. Zeitschrift für mathematische Logik und Grundlagen der Mathematik **38**, 179–187 (1992)
9. del Cerro, L.F., Orłowska, E.: DAL—a logic for data analysis. Theoret. Comput. Sci. **36**, 251–264 (1985)
10. Fisher, M., Ladner, R.: Propositional dynamic logic of regular programs. J. Comput. Syst. Sci. **18**, 194–211 (1979)
11. Frias, M.: Fork Algebras in Algebra, Logic and Computer Science. World Scientific, River Edge (2002)
12. Frias, M., Baum, G., Haeberer, A.: Fork algebras in algebra, logic and computer science. Fund. Inf. **32**, 1–25 (1997)
13. Frias, M., Veloso, P., Baum, G.: Fork algebras: past, present and future. J. Relat. Methods Comput. Sci. **1**, 181–216 (2004)
14. Gargov, G., Passy, S.: A note on Boolean modal logic. In: Petkov, P. (ed.) Mathematical Logic, pp. 299–309. Plenum Press, New York (1990)
15. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press, Cambridge (2000)

16. Kozen, D., Parikh, R.: An elementary proof of the completeness of PDL. Theoret. Comput. Sci. **14**, 113–118 (1981)
17. Mikulás, S.: Complete calculus for conjugated arrow logic. In: Marx, M., Pólos, L., Masuch, M. (eds.) Arrow Logic and Multi-Modal Logic, pp. 125–139. CSLI, Stanford (1996)
18. Mirkowska, G.: PAL—propositional algorithmic logic. Fund. Inf. **4**, 675–760 (1981)
19. Tarski, A.: On the calculus of relations. J. Symb. Logic **6**, 73–89 (1941)

http://www.springer.com/978-3-319-73578-8

Dynamic Logic. New Trends and Applications
First International Workshop, DALI 2017, Brasilia, Brazil,
September 23-24, 2017, Proceedings
Madeira, A.; Benevides, M. (Eds.)
2018, IX, 201 p. 25 illus., Softcover
ISBN: 978-3-319-73578-8