Ockhamist Propositional Dynamic Logic: a natural link between PDL and CTL

Philippe Balbiani, Emiliano Lorini

Ockhamist Propositional Dynamic Logic: A Natural Link between PDL and CTL*

Philippe Balbiani and Emiliano Lorini

Université de Toulouse, IRIT-CNRS, France

Abstract. We present a new logic called Ockhamist Propositional Dynamic Logic, OPDL, which provides a natural link between PDL and CTL*. We show that both PDL and CTL* can be polynomially embedded into OPDL in a rather simple and direct way. More generally, the semantics on which OPDL is based provides a unifying framework for making the dynamic logic family and the temporal logic family converge in a single logical framework. Decidability of the satisfiability problem for OPDL is studied in the paper.

1 Introduction

Different logical systems are traditionally used in theoretical computer science and in artificial intelligence for the verification of programs and for modelling reactive systems and multi-agent systems. Among them we should mention Propositional Dynamic Logic PDL [12], Propositional Linear Temporal Logic PLTL [20], Computation Tree Logic CTL [11], Full Computation Tree Logic CTL* [22] and Alternating-time Temporal Logic ATL [1]. Some relationships between these different logical systems have been studied. For instance, it is well-known that PLTL and CTL are fragments of CTL* and that CTL is a fragment of ATL [13]. However, at the current stage, the general picture remains incomplete. For example, it is clear (and well-known) that the logic of programs PDL can express properties that Full Computation Tree Logic CTL* cannot and vice-versa. Moreover, there are no clear relationships between PDL and logics of strategic reasoning such as ATL. More precisely, it is not known whether there exists natural embeddings of PDL into ATL or of ATL in PDL. Even more importantly, there is still no logical system that can be said to be more general than the others. For instance, there is no logic that embeds in a *natural* and *simple* way both PDL and CTL*. Indeed, although there exist some logics that embed both PDL and CTL*, they do it in a rather complicated and unnatural way. For example, it is well-known that PDL and CTL* can be embedded in modal μ-calculus. However, although the embedding of PDL into modal μ-calculus is simple and direct, the embedding of CTL* into modal μ-calculus is rather complicated and doubly exponential in the length of the input formula [7]. Another logic that links PDL with CTL* is the extension of PDL with a repetition construct (PDL-Δ) by [26]. But again, the embedding of CTL* into PDL-Δ too

is rather complicated and doubly exponential in the length of the input formula [29].[1] For this reason, a challenge arises of making the previous competing logical systems converge into a single logical system. The aim of this paper is to make a step into this direction by proposing an Ockhamist variant of Propositional Dynamic Logic, OPDL, that provides a natural link between PDL and CTL*. Specifically, we show that both PDL and CTL* can be polynomially embedded into OPDL in a rather simple and direct way. More generally, the Ockhamist semantics on which OPDL is based provides a unifying framework for making the dynamic logic family and the temporal logic family converge into a single logical framework. Ockhamist semantics for temporal logic have been widely studied in the 80ies and in the 90ies [27,30,5]. The logic of agency STIT (the logic of "seeing to it that") by Belnap *et al.* [4] is based on such semantics. According to the Ockhamist conception of time (also called *indeterminist actualist*, see [30]) the truth of statements is evaluated with respect to a moment and to a particular *actual* linear history passing through that moment, and the temporal operators are relativized to the actual history of the evaluation.

The rest of the paper is organized as follows. We first present the syntax and the semantics of OPDL and provide a decidability result for this logic (Section 2). Then, we discuss, in Section 3, about the relationship of OPDL with PDL and CTL* (Section 3). In particular, we provide polynomial reductions of PDL and CTL* to OPDL. In Section 4 we present a variant of OPDL whose semantics is based on the notion of labeled transition system (LTS). In Section 5 we conclude by discussing some perspectives for future work.[2]

2 Ockhamist Propositional Dynamic Logic

The distinction between the 'Ockhamist' semantics and the 'Peircean' semantics for branching-time temporal logic was proposed by Prior in his seminal work on the logic of time [21] (see also [27]). According to the 'Peircean' view the truth of a temporal formula should be evaluated with respect either to some history or all histories starting in a given state. In the 'Ockhamist' semantics for branching time a notion of *actual* course of events is given. In particular, according to the 'Ockhamist' view, the truth of a temporal formula should be evaluated with respect to a particular *actual* history starting in a given state. While the branching-time temporal logic CTL* is compatible with the Ockhamist conception of time, the semantics for PDL in terms of labelled transition systems is closer to the Peircean view than to the Ockhamist view since it does not consider a notion of actual history or actual path in a transition system. The logic OPDL can be conceived as a variant of the logic of programs PDL based on the

[1] It is worth noting that Axelsson et al. [2] have recently studied generic extensions of CTL in which temporal operators are parameterized with different kinds of formal languages recognized by different classes of automata (*e.g.*, regular languages, visibly pushdown and context-free languages). They compare the expressive power of these extensions of PDL to CTL, PDL and extensions of PDL such as PDL-Δ. However, they also show that CTL* cannot be embedded in any of these extensions of CTL, as the property of fairness is expressible in CTL* but is not expressible is any of these logics (see [2, Theorem 4.3]).

[2] An extended version of this paper containing detailed proofs is available at [3].

Ockhamist view of time. Specifically, OPDL is a variant of PDL in which the truth of a formula is evaluated with respect to a given actual history. The syntax and the semantics of this logic are presented in Sections 2.1 and 2.2.

2.1 Syntax

Assume a countable set $Prop$ of atomic propositions (with typical members denoted $p, q, \ldots$) and a countable set Atm of atomic programs (or atomic actions) (with typical members denoted $a, b, \ldots$). Let $2^{Atm*} = 2^{Atm} \setminus \{\emptyset\}$. The language $\mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ of OPDL consists of a set Prg of programs and a set Fml of formulae. It is defined as follows:

$$\begin{array}{l} Prg: \ \pi ::= a \mid \equiv \mid (\pi_1; \pi_2) \mid (\pi_1 \cup \pi_2) \mid \pi^* \mid \varphi? \\ Fml: \varphi ::= p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid [\![\pi]\!]\varphi \end{array}$$

where p ranges over $Prop$ and a ranges over Atm. We adopt the standard definitions for the remaining Boolean operations. The dual $\langle\!\langle \pi \rangle\!\rangle$ of the operator $[\![\pi]\!]$ is defined in the expected way: $\langle\!\langle \pi \rangle\!\rangle\varphi \stackrel{\mathtt{def}}{=} \neg[\![\pi]\!]\neg\varphi$. We follow the usual rules for omission of the parentheses. Given a formula φ, let $FL(\varphi)$ denote its Fischer-Ladner closure. See [12, Chapter 6] for details. It is a well-known fact that $card(FL(\varphi))$ is linear in the length of φ.

Complex programs of sequential composition ($\pi_1; \pi_2$), non-deterministic choice ($\pi_1 \cup \pi_2$), iteration (π^*) and test ($\varphi?$) are built from atomic programs in Atm, from the special program $\equiv$ and from formulae in Fml. The special program $\equiv$ allows to move from a history to an alternative history passing through the same moment. The behavior of this program will become clearer in Section 2.2 when presenting the OPDL semantics.

The formula $[\![\pi]\!]\varphi$ has to be read "φ will be true at the end of all possible executions of program π" whereas $\langle\!\langle \pi \rangle\!\rangle\varphi$ has to be read "φ will be true at the end of some possible execution of program π". As it is assumed that atomic programs in Atm are linear (*i.e.*, all atomic programs in Atm occurring at a given state lead to the same successor state), $[\![a]\!]\varphi$ can also be read "if the atomic program a occurs, φ will be true afterwards". Indeed, from the assumption of linearity, it follows that atomic programs in Atm are deterministic (*i.e.*, there is at most one possible execution of an atomic program a at a given state). Finally, the formula $[\![\equiv]\!]\varphi$ has to be read "φ is true in all histories passing through the current moment" or, more shortly, "φ is necessarily true in the current moment".

2.2 Semantics

OPDL frames are structures with two dimensions: a vertical dimension corresponding to the concept of history, a horizontal dimension corresponding to the concept of moment.

Definition 1 (OPDL frame). *An OPDL frame is a tuple $F = (W, \mathcal{Q}, \mathcal{L}, \mathcal{R}_{\equiv})$ where:*

- *W is a nonempty set of states (or worlds),*
- *$\mathcal{Q}$ is a partial function $\mathcal{Q} : W \longrightarrow W$,*

- *$\mathcal{L}$ is a mapping $\mathcal{L} : \mathcal{Z} \longrightarrow 2^{Atm*}$ from state transitions to non-empty sets of atomic programs, $\mathcal{Z} = \{(w, v) \mid w, v \in W$ and $\mathcal{Q}(w) = v\}$ being the transition relation induced by the successor state function $\mathcal{Q}$,*
- *$\mathcal{R}_{\equiv} \subseteq W \times W$ is an equivalence relation between states in W such that for all $w, v, u \in W$:*
  - ***(C1)*** *if $\mathcal{Q}(w) = v$ and $(v, u) \in \mathcal{R}_{\equiv}$ then there is $z \in W$ such that $(w, z) \in \mathcal{R}_{\equiv}$ and $\mathcal{Q}(z) = u$ and $\mathcal{L}(z, u) = \mathcal{L}(w, v)$.*

For every $w, v \in W$, $\mathcal{Q}(w) = v$ means that v is the successor state of w. If $\mathcal{Q}(w) = v$ then we also say that w is a predecessor of v. If $\mathcal{L}(w, v) = \{a, b\}$, then the actions a and b are responsible for the transition from the state w to the state v. In other words, the function $\mathcal{L}$ labels every state transition with a set of atomic actions (*viz.* the actions that are responsible for the transition). The assumption that the set $\mathcal{L}(w, v)$ should be non-empty means that every state transition is due to the execution of at least one atomic action.

$\mathcal{R}_{\equiv}$-equivalence classes are called *moments*. If w and v belong to the same moment then they are called *alternatives*. A maximal sequence of states according to the transition relation $\mathcal{Z}$ starting at a given state w is called *history starting in* w. If w and v belong to the same moment, then the history starting in w and the history starting in v are alternative histories (*viz.* histories starting at the same moment).

Constraint (**C1**) corresponds to what in Ockhamist semantics is called property of *weak diagram completion* [30]. This means that if two worlds v and u are in the same moment and world w is a predecessor of v then, there exists a world z such that (i) w and z are in the same moment, (ii) u is the successor of z, (iii) the transition from w to v and the transition from z to u are labeled with the same set of action names.

Figure 1 is an example of OPDL frame. The $\mathcal{R}_{\equiv}$-equivalences classes $\{w_1, w_2, w_3, w_4\}$, $\{w_5, w_6\}$, $\{w_7, w_8\}$, $\{w_9\}$, $\{w_{10}\}$, $\{w_{11}\}$, $\{w_{12}\}$, $\{w_{13}\}$, $\{w_{14}\}$, $\{w_{15}\}$ and $\{w_{16}\}$ are the moments. The sequences of states (w_1, w_5, w_9, w_{13}), $(w_2, w_6, w_{10}, w_{14})$, $(w_3, w_7, w_{11}, w_{15})$ and $(w_4, w_8, w_{12}, w_{16})$ are the alternative histories starting at the same moment $\{w_1, w_2, w_3, w_4\}$. Actions a and c are responsible for the transition from the state w_1 to the state w_5 and, because of Constraint (**C1**), actions a and c are also responsible for the transition from the state w_2 to the state w_6. Moreover, actions b and c are responsible for the transition from the state w_3 to the state w_7 and, because of Constraint (**C1**), actions b and c are also responsible for the transition from the state w_4 to the state w_8.

Definition 2 (Atomic transitions). *Given an OPDL frame $F = (W, \mathcal{Q}, \mathcal{L}, \mathcal{R}_{\equiv})$ and an atomic program $a \in Atm$, let*

$$\mathcal{R}_a = \{(w, v) \mid \mathcal{Q}(w) = v \text{ and } a \in \mathcal{L}(w, v)\}$$

be the set of a-transitions in the frame F.

An OPDL model is an OPDL frame supplemented with a valuation function mapping each state to the set of propositional atoms which are true in it, under the assumption that two states belonging to the same moment agree on the atoms. More precisely:

Definition 3 (OPDL model). *An OPDL model is a tuple $M = (W, \mathcal{Q}, \mathcal{L}, \mathcal{R}_{\equiv}, \mathcal{V})$ where:*

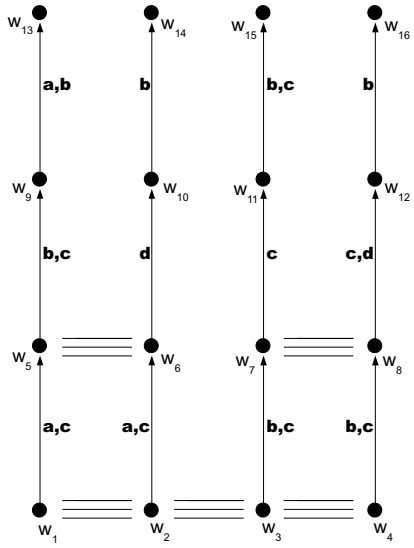

Fig. 1. An OPDL frame

- $(W, \mathcal{Q}, \mathcal{L}, \mathcal{R}_{\equiv})$ *is a* OPDL *frame and*
- $\mathcal{V} : W \longrightarrow 2^{Prop}$ *is a valuation function for atomic propositions such that for all* $w, v \in W$*:*
  (C2) *if* $(w, v) \in \mathcal{R}_{\equiv}$ *then* $\mathcal{V}(w) = \mathcal{V}(v)$.

The truth of a OPDL formula is evaluated with respect to a world w in an OPDL model M.

Definition 4 (π-transitions and truth conditions). *Let* $M = (W, \mathcal{Q}, \mathcal{L}, \mathcal{R}_{\equiv}, \mathcal{V})$ *be an* OPDL *model. Given a program* π*, let us define a binary relation* $\mathcal{R}_\pi$ *on* W *with* $(w, v) \in \mathcal{R}_\pi$ *(or* $w\mathcal{R}_\pi v$*) meaning that* v *is accessible from* w *by performing program* π*. Let us also define a binary relation* $\models$ *between worlds in* M *and formulae with* $M, w \models \varphi$ *meaning that formula* φ *is true at* w *in* M*. The rules inductively defining* $\mathcal{R}_\pi$ *and* $\models$ *are:*

$$\begin{aligned}
\mathcal{R}_{\pi_1;\pi_2} &= \mathcal{R}_{\pi_1} \circ \mathcal{R}_{\pi_2} \\
\mathcal{R}_{\pi_1 \cup \pi_2} &= \mathcal{R}_{\pi_1} \cup \mathcal{R}_{\pi_2} \\
\mathcal{R}_{\pi^*} &= (\mathcal{R}_\pi)^* \\
\mathcal{R}_{\varphi?} &= \{(w, w) \mid w \in W \text{ and } M, w \models \varphi\}
\end{aligned}$$

and

$$\begin{aligned}
M, w \models p &\iff p \in \mathcal{V}(w); \\
M, w \models \neg\varphi &\iff M, w \not\models \varphi; \\
M, w \models \varphi \wedge \psi &\iff M, w \models \varphi \text{ AND } M, w \models \psi; \\
M, w \models [\![\pi]\!]\varphi &\iff \forall v \in \mathcal{R}_\pi(w) : M, v \models \varphi
\end{aligned}$$

with $\mathcal{R}_\pi(w) = \{v \in W \mid (w, v) \in \mathcal{R}_\pi\}$.

An OPDL formula φ is said to be OPDL valid, denoted by $\models_{\mathsf{OPDL}} \varphi$, if and only if φ is true in all OPDL models (*i.e.*, for every OPDL model M and for every world w in M, we have $M, w \models \varphi$). An OPDL formula φ is said to be OPDL satisfiable if and only if $\neg\varphi$ is not OPDL valid.

OPDL formulae can also be interpreted over standard Kripke structures.

Definition 5 (Kripke OPDL model). *A Kripke OPDL model is a tuple* $M = (W, \{\mathcal{R}_a \mid a \in Atm\}, \mathcal{R}_\equiv, \mathcal{V})$ *where:*

- W *is a set of states (or worlds),*
- $\mathcal{R}_\equiv$ *is an equivalence relation on* W *and all* $\mathcal{R}_a$ *are binary relations on* W *satisfying the following two constraints for all* $w, v, u \in W$*:*
  *(**C1***)* *if* $(w, v) \in \mathcal{R}_\mathsf{X}$ *and* $(w, u) \in \mathcal{R}_\mathsf{X}$ *then* $u = v$*,*
  *(**C2***)* *if* $(w, v) \in \mathcal{R}_\mathsf{X}$ *and* $(v, u) \in \mathcal{R}_\equiv$ *then there is* $z \in W$ *such that* $(w, z) \in \mathcal{R}_\equiv$ *and for all* $a \in Atm$*,* $(w, v) \in \mathcal{R}_a$ *if and only if* $(z, u) \in \mathcal{R}_a$*,*
  with $\mathcal{R}_\mathsf{X} = \bigcup_{a \in Atm} \mathcal{R}_a$*,*
- $\mathcal{V} : W \longrightarrow 2^{Prop}$ *is a valuation function for atomic propositions such that for all* $w, v \in W$*:*
  *(**C3***)* *if* $(w, v) \in \mathcal{R}_\equiv$ *then* $\mathcal{V}(w) = \mathcal{V}(v)$.

Constraints (**C2***) and (**C3***) are respectively the counterparts of Constraint (**C1**) and Constraint (**C2**) in the definition of an OPDL model. Constraint (**C1***) forces the successor relation $\mathcal{R}_\mathsf{X}$ to be deterministic (*i.e.*, every state has at most one successor).

As stated by the following proposition, the notion of satisfiability with respect to the class of OPDL models is equivalent to the notion of satisfiability with respect to the class of Kripke OPDL models.

Proposition 1. *Let* φ *be an OPDL formula. Then, there exists an OPDL model which satisfies* φ *if and only if there exists a Kripke OPDL model which satisfies* φ*.*

We shall say that φ is a global logical consequence of a finite set of global axioms $\Gamma = \{\chi_1, \ldots, \chi_n\}$, denoted by $\Gamma \models_{\mathsf{OPDL}} \varphi$, if and only if for every OPDL model M, if Γ is true in M (*i.e.*, for every world w in M, we have $M, w \models \chi_1 \wedge \ldots \wedge \chi_n$) then φ is true in M too (*i.e.*, for every world w in M, we have $M, w \models \varphi$).

As the following proposition highlights, when the set of atomic programs Atm is finite, the problem of logical consequence in OPDL with a finite set of global axioms is reducible to the validity problem for OPDL formulae.

Proposition 2. *Let* $\Gamma = \{\chi_1, \ldots, \chi_n\}$ *be a finite set of OPDL formulae. If* Atm *is finite,* $\Gamma \models_{\mathsf{OPDL}} \varphi$ *if and only if* $\models_{\mathsf{OPDL}} [\![\mathbf{any}^*]\!](\chi_1 \wedge \ldots \wedge \chi_n) \rightarrow \varphi$ *with* $\mathbf{any} \stackrel{\mathtt{def}}{=} (\bigcup_{a \in Atm} \cup \equiv)$.

The model checking problem for OPDL is the following decision problem: given a finite OPDL model M and an OPDL formula φ, is there a world in M such that $M, w \models \varphi$? With finite OPDL model, we mean a OPDL model $M = (W, \mathcal{Q}, \mathcal{L}, \mathcal{R}_\equiv, \mathcal{V})$ that satisfies the following three conditions: (1) W is *finite*; (2) $\mathcal{L}$ associates to every transition $(w, v) \in \mathcal{Z} = \{(w, v) \mid w, v \in W \text{ and } \mathcal{Q}(w) = v\}$ a non-empty *finite* set of atomic

actions in Atm; (3) $\mathcal{V}$ associates to every world $w \in W$ a *finite* set of atomic formulas in $Prop$. In order to determine whether there exists a world w in M such that $M, w \models \varphi$, we can use the model checking algorithm for PDL showing that the model checking problem for PDL is PTIME-complete with respect to the size of the input model and the input formula. It follows that the model checking problem for OPDL is PTIME-complete too with respect to $size(M) + size(\varphi)$.

2.3 Decidability of OPDL

Using the "mosaic method", a technique used in algebraic logic [18] to prove the decidability of equational theories, we will prove the decidability of SAT, the following decision problem: determine whether a given OPDL formula φ is satisfiable with respect to the class of OPDL models.

Theorem 1. *SAT is decidable.*

Proof (Sketch). Let φ be a formula. In order to simplify the proof, we assume that at most one atomic action, namely a, occurs in φ. A type for φ is a subset t of $FL(\varphi)$. It is normal iff it satisfies the conditions of atomicity considered in [14, Definition 2.2]. A group for φ is a finite set G of normal types for φ. A mosaic for φ is a finite set M of groups for φ. It is normal iff it satisfies the following conditions: **(i)** if $p \in FL(\varphi)$ then for all $G \in M$, for all $t \in G$, if $p \in t$ then for all $H \in M$, for all $u \in H$, $p \in u$; **(ii)** if $[\![\equiv]\!]\psi \in FL(\varphi)$ then for all $G \in M$, for all $t \in G$, if $[\![\equiv]\!]\psi \in t$ then for all $H \in M$, for all $u \in H$, $\psi \in u$; **(iii)** if $\neg[\![\equiv]\!]\psi \in FL(\varphi)$ then for all $G \in M$, for all $t \in G$, if $\neg[\![\equiv]\!]\psi \in t$ then there exists $H \in M$, there exists $u \in H$ such that $\neg\psi \in u$. A system for φ is a finite set S of normal mosaics for φ. A context for S is a structure of the form (M, G, t) where $M \in S$, $G \in M$ and $t \in G$. Obviously, there exists finitely many types, groups, mosaics and systems for φ. Since the normality conditions for types and mosaics are decidable, the set of all contexts can be computed. Let $\Sigma = \{a, \equiv\} \cup \{\psi?\colon \psi \in FL(\varphi)\}$. For all $\alpha \in \Sigma$, we define the transition relation $\longrightarrow^S_\alpha$ between contexts for S as follows: $(M, G, t) \longrightarrow^S_\alpha (N, H, u)$ iff one of the following conditions is satisfied: **(i)** $\alpha = a$ and there exists a bijection $f : G \to N$ such that **(a)** $f(t) = u$, **(b)** if $[\![a]\!]\psi \in FL(\varphi)$ then for all $v \in G$, if $[\![a]\!]\psi \in v$ then $\psi \in f(v)$, **(c)** if $\neg[\![a]\!]\psi \in FL(\varphi)$ then for all $v \in G$, if $\neg[\![a]\!]\psi \in v$ then $\neg\psi \in f(v)$; **(ii)** $\alpha = \equiv$ and $M = N$; **(iii)** $\alpha = \psi?$, $M = N$, $G = H$, $t = u$ and $\psi \in u$. For all programs π, we inductively define the transition relation $\longrightarrow^S_\pi$ between contexts for S as follows: $\longrightarrow^S_{\pi_1;\pi_2} = \longrightarrow^S_{\pi_1} \circ \longrightarrow^S_{\pi_2}$, $\longrightarrow^S_{\pi_1 \cup \pi_2} = \longrightarrow^S_{\pi_1} \cup \longrightarrow^S_{\pi_2}$, $\longrightarrow^S_{\pi^\star} = (\longrightarrow^S_\pi)^\star$. Since the set of all contexts can be computed, the transition relations $\longrightarrow^S_\pi$ are all decidable. A system S for φ is said to be saturated iff it satisfies the following condition: if $\neg[\![\pi]\!]\psi \in FL(\varphi)$ then for all contexts (M, G, t) for S, if $\neg[\![\pi]\!]\psi \in t$ then there exists a context (N, H, u) for S such that $(M, G, t) \longrightarrow^S_\pi (N, H, u)$ and $\neg\psi \in u$. Since the transition relations $\longrightarrow^S_\pi$ are all decidable, checking the saturation of a given system for φ is decidable. The proof of the decidability of SAT proceeds in two steps. First, in Proposition 3, we prove that φ is satisfiable iff there exists a saturated system S for φ and there exists a context (M, G, t) in S such that $\varphi \in t$. Second, in Proposition 4, we prove the decidability of the decision problem SYS defined as follows: determine whether there exists a saturated system S for a given formula φ and there exists a context (M, G, t) in S such that $\varphi \in t$.

Proposition 3. *Let φ be a formula. The following conditions are equivalent:* **(i)** *φ is satisfiable;* **(ii)** *there exists a saturated system S for φ, there exists a context (M, G, t) in S such that $\varphi \in t$.*

Proposition 4. *SYS is decidable.*

As a result, SAT is decidable. □

3 Relationships between OPDL, PDL and CTL*

In this section we study the relationships between OPDL and PDL, and between OPDL and CTL*. In particular, we provide a polynomial embedding of PDL into OPDL and a polynomial embedding of CTL* into OPDL.

3.1 Relationships between OPDL and PDL

Propositional Dynamic Logic PDL [15] is the well-known logic of programs. Again assume the countable set of atomic propositions $Prop = \{p, q, \ldots\}$ and the countable set of atomic programs $Atm = \{a, b, \ldots\}$. The language $\mathcal{L}_{\mathsf{PDL}}(Prop, Atm)$ of PDL is defined by the following grammar in Backus-Naur Form (BNF):

$$Prg: \ \pi ::= a \mid (\pi_1; \pi_2) \mid (\pi_1 \cup \pi_2) \mid \pi^* \mid \varphi?$$
$$Fml: \varphi ::= p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid [\pi]\varphi$$

where p ranges over $Prop$ and a ranges over Atm.

PDL models are nothing but labeled transition systems, *i.e.*, transition systems where transitions between states are labeled with atomic programs.

Definition 6. *PDL models are tuples $M = (W, \{\mathcal{R}_a \mid a \in Atm\}, \mathcal{V})$ where:*

- *W is a nonempty set of possible worlds or states;*
- *$\{\mathcal{R}_a \mid a \in Atm\}$ is a set of binary relations on W;*
- *$\mathcal{V} : W \longrightarrow 2^{Prop}$ is a valuation function.*

The accessibility relations for atomic programs are generalized to complex programs in the usual way (see Definition 2).

The truth conditions of PDL formulae are standard for the Boolean constructions plus the following one for the dynamic operators $[\pi]$:

$$M, w \models [\pi]\varphi \iff \forall v \in \mathcal{R}_\pi(w) : M, v \models \varphi$$

A PDL formula φ is said to be PDL valid if and only if φ is true in all PDL models.

We can embed PDL in OPDL. Consider the following polynomial translation $tr_1 : \mathcal{L}_{\mathsf{PDL}}(Prop, Atm) \longrightarrow \mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ from the language of PDL to the OPDL language.

$$\begin{array}{ll} tr_1(p) & = p \ \text{ for all } p \in Prop \\ tr_1(\neg\varphi) & = \neg tr_1(\varphi) \\ tr_1(\varphi_1 \wedge \varphi_2) & = tr_1(\varphi_1) \wedge tr_1(\varphi_2) \\ tr_1([\pi]\varphi) & = [\![tr_2(\pi)]\!] tr_1(\varphi) \end{array}$$

where

$$\begin{array}{ll} tr_2(a) & = \equiv; a \text{ for all } a \in Atm \\ tr_2(\pi_1;\pi_2) & = tr_2(\pi_1); tr_2(\pi_2) \\ tr_2(\pi_1 \cup \pi_2) & = tr_2(\pi_1) \cup tr_2(\pi_2) \\ tr_2(\pi^*) & = (tr_2(\pi))^* \\ tr_2(\varphi?) & = tr_1(\varphi)? \end{array}$$

As the following theorem shows, the preceding translation is a correct embedding.

Theorem 2. *Let φ be a PDL formula. φ is PDL valid if and only if $tr_1(\varphi)$ is OPDL valid.*

3.2 Relationships between OPDL and CTL*

Full Computation Tree Logic CTL* was first described in [10,9] as an extension of Computation Tree Logic CTL [6] and Propositional Linear Temporal Logic PLTL [20]. The language of CTL* is built recursively from the atomic propositions using the temporal operators of PLTL, and the existential path switching operator of CTL as well as classical connectives.

Again assume the countable set of atomic propositions $Prop = \{p, q, \ldots\}$. The language $\mathcal{L}_{\mathsf{CTL}^*}(Prop)$ of CTL* is defined by the following grammar in Backus-Naur Form (BNF):

$$\varphi ::= p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \mathsf{X}\varphi \mid \varphi\,\mathcal{U}\,\psi \mid \mathsf{A}\varphi$$

where p ranges over $Prop$. The constructs X and $\mathcal{U}$ are respectively the operators *next* and *until* of PLTL, the formulas $\mathsf{X}\varphi$ and $\varphi\,\mathcal{U}\,\psi$ being respectively read "φ will be true in the next state along the current path" and "ψ will be true at some point in the future along the current path and φ has to hold until ψ". These two operators can be used to express other kinds of temporal notions such as *eventually* $\mathsf{F}\psi \stackrel{\mathtt{def}}{=} \top\,\mathcal{U}\,\psi$, *henceforth* $\mathsf{G}\psi \stackrel{\mathtt{def}}{=} \neg\mathsf{F}\neg\psi$ and *before* $\varphi\,\mathcal{B}\,\psi \stackrel{\mathtt{def}}{=} \neg(\neg\varphi\,\mathcal{U}\,\psi)$. The construct A is a modal operator quantifying over possible paths, the formula $\mathsf{A}\varphi$ being read "φ is true in all possible paths". The existential path-quantifier operator E, is defined by $\mathsf{E}\varphi \stackrel{\mathtt{def}}{=} \neg\mathsf{A}\neg\varphi$.

Different semantics for CTL* have been given in the literature. One of this semantics is based on the notion of Ockhamist structure. Here we mainly follow the presentation of the Ockhamist semantics for CTL* given by Reynolds [22] who introduces a special kind of Ockhamist structures called $(\mathbb{N} \times W)$ structures.

Definition 7. *A $(\mathbb{N} \times W)$ structure is a tuple $(W, \sim, g)$ where:*

- *W is a set of points;*
- *$\sim$ is an equivalence relation over $\mathbb{N} \times W$ such that for all $w, v \in W$ and for all $n, m \in \mathbb{N}$:*
  - ***(S1)*** *if $(n, w) \sim (m, v)$ then $n = m$,*
  - ***(S2)*** *if $(n, w) \sim (n, v)$ and $m < n$ then $(m, w) \sim (m, v)$,*
  - ***(S3)*** *$(0, w) \sim (0, v)$;*

– $g : \mathbb{N} \times W \longrightarrow 2^{Prop}$ *is a valuation function mapping each integer and each point into a set of atoms such that for all* $w, v \in W$ *and for all* $n \in \mathbb{N}$*:*
(S4) *if* $(n, w) \sim (n, v)$ *then* $g(n, w) = g(n, v)$.

Given a $(\mathbb{N} \times W)$ structure $(W, \sim, g)$ and a CTL^* formula φ, $(W, \sim, g), (n, w) \models \varphi$ means that φ is true at the index (n, w) in the $(\mathbb{N} \times W)$ structure $(W, \sim, g)$. The rules defining the truth conditions of CTL^* formulae are inductively defined as follows:

$$
\begin{aligned}
(W, \sim, g), (n, w) \models p &\iff p \in g(n, w);\\
(W, \sim, g), (n, w) \models \neg\varphi &\iff (W, \sim, g), (n, w) \not\models \varphi;\\
(W, \sim, g), (n, w) \models \varphi_1 \wedge \varphi_2 &\iff (W, \sim, g), (n, w) \models \varphi_1 \text{ AND } (W, \sim, g), (n, w) \models \varphi_2;\\
(W, \sim, g), (n, w) \models \mathsf{X}\varphi &\iff (W, \sim, g), (n + 1, w) \models \varphi\\
(W, \sim, g), (n, w) \models \varphi\, \mathcal{U}\, \psi &\iff \exists m \in \mathbb{N} : m \geq n \text{ AND } (W, \sim, g), (m, w) \models \psi \text{ AND}\\
&\qquad\qquad \forall k \in \mathbb{N} : \text{IF } n \leq k < m \text{ THEN } (W, \sim, g), (k, w) \models \varphi\\
(W, \sim, g), (n, w) \models \mathsf{A}\varphi &\iff \forall v \in W : \text{IF } (n, w) \sim (n, v) \text{ THEN } (W, \sim, g), (n, v) \models \varphi
\end{aligned}
$$

As shown by Reynolds [22] the CTL^* semantics in terms of $(\mathbb{N} \times W)$ structures is equivalent to the CTL^* semantics in terms of *bundled trees*. However, it is more general than the common CTL^* semantics in terms of $\mathcal{R}$*-generable models* used by [9], *i.e.*, Kripke structures with states, a total accessibility relation $\mathcal{R}$ between them and the set of all paths which arise by moving from state to state along the accessibility relation. The difference between the CTL^* semantics in terms of bundled trees and the CTL^* semantics in terms of R-generable models is that the latter quantifies over all paths induced by the relation R whereas the former quantifies over a bundle of paths. Although this bundle is suffix and fusion closed, it does not need to be be limit closed. For example, it may be the case that all paths include a right branch even though at every world there is a path where the next branch goes left, which violates the limit closure property. In order to distinguish full computation tree logic interpreted over R-generable models and full computation tree logic interpreted over bundled trees (and equivalently over $(\mathbb{N} \times W)$ structures), some authors (see, *e.g.*, [23,16,17]) use the term CTL^* to indicate the former logic and the term BCTL^* (bundled CTL^*) to indicate the latter (in [25] it is called $\forall LTFC$).

It is well-known that CTL^* interpreted over R-generable models is 2-EXPTIME complete: [10] provides a doubly exponential automaton based satisfiability checker, and [28] gives a lowerbound. As pointed by [17], an argument for the 2-EXPTIME hardness of the satisfiability problem could also be made for CTL^* interpreted over bundled trees in a way similar to the argument for CTL^* interpreted over R-generable models. Therefore, as the CTL^* semantics in terms of bundled trees is equivalent to the CTL^* semantics in terms of $(\mathbb{N} \times W)$ structures [22], it follows that the satisfiability problem for CTL^* interpreted over $(\mathbb{N} \times W)$ structures is also 2-EXPTIME hard. Therefore, CTL^* interpreted over bundled trees (or over $(\mathbb{N} \times W)$ structures) is not easier to deal with than CTL^* interpreted over R-generable models. However, one interesting aspect of the former kind of CTL^* is that it is closely connected to Ockhamist temporal logics studied in the field of philosophical logic [30]. Moreover, one might argue that reasoning in BCTL^* is relatively easier than reasoning in CTL^*. For example the specification for the tableau method for BCTL^* proposed by [23] was much simpler than the CTL^* tableau that originated from it [24].

Consider the following translation $tr_3 : \mathcal{L}_{\mathsf{CTL}^*}(Prop) \longrightarrow \mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ from the language of CTL^* to the OPDL language where x is an arbitrary atomic program in Atm:

$$\begin{array}{ll} tr_3(p) & = p \text{ for all } p \in Prop \\ tr_3(\neg\varphi) & = \neg tr_3(\varphi) \\ tr_3(\varphi_1 \wedge \varphi_2) & = tr_3(\varphi_1) \wedge tr_3(\varphi_2) \\ tr_3(\mathsf{X}\varphi) & = \langle\!\langle x \rangle\!\rangle tr_3(\varphi) \\ tr_3(\varphi\, \mathcal{U}\, \psi) & = \langle\!\langle (tr_3(\varphi)?; x)^* \rangle\!\rangle tr_3(\psi) \\ tr_3(\mathsf{A}\varphi) & = [\![\equiv]\!] tr_3(\varphi) \end{array}$$

The preceding translation is polynomial and, as the following theorem shows, it provides an embedding of the variant of CTL^* interpreted over $(\mathbb{N} \times W)$ structures into OPDL.

Theorem 3. *Let φ be a CTL^* formula. φ is valid with respect to the class of $(\mathbb{N} \times W)$ structures if and only if $\{\langle\!\langle x \rangle\!\rangle \top\} \models_{\mathsf{OPDL}} tr_3(\varphi)$.*

From Theorem 3 and Proposition 2 in Section 2.2, it follows that the satisfiability problem in the variant of CTL^* interpreted over $(\mathbb{N} \times W)$ structures can be reduced to the satisfiability problem in OPDL with a finite number of atomic programs.

Corollary 1. *Let φ be a CTL^* formula and let Atm be finite. φ is valid with respect to the class of $(\mathbb{N} \times W)$ structures if and only if $\models_{\mathsf{OPDL}} [\![\mathbf{any}^*]\!] \langle\!\langle x \rangle\!\rangle \top \rightarrow tr_3(\varphi)$.*

Since the satisfiability problem of $(\mathbb{N} \times W)$ structures is 2-EXPTIME-hard (see above), the preceding polynomial embedding of CTL^* into OPDL provides an argument for the 2-EXPTIME-hardness of the satisfiability problem of our logic OPDL.

3.3 Relationships with Other Logics: Discussion

The logic OPDL has interesting connections with other logical systems proposed in the field of theoretical computer science such as propositional linear time temporal logic PLTL and Nishimura's combination of PDL and PLTL (call it, NL) [19]. As for PLTL, in the previous Section 3.2 we have provided a polynomial embedding of CTL^* into OPDL. As PLTL is nothing but the fragment of CTL^* without the path quantifier operator A, the translation tr_3 also provides a polynomial embedding of PLTL into OPDL. As for NL, we just need to put together the translation tr_1 from PDL to OPDL given in Section 3.1 and the translation from PLTL to OPDL in order to provide a polynomial reduction of Nishimura's logic NL to OPDL. Another logic that is related with OPDL is ACTL^*, the action based version of CTL^* proposed by [8]. ACTL^* extends CTL^* with temporal operators X_a indexed by atomic programs a in the set of atomic programs Atm. The ACTL^* formula $\mathsf{X}_a\varphi$ has to be read "the next transition is labeled with the atomic program a and φ will be true in the next state along the current path". By adding the following item to the preceding translation tr_3 from CTL^* to OPDL, we get a polynomial embedding of ACTL^* into OPDL:

$$tr_3(\mathsf{X}_a\varphi) = \langle\!\langle a \rangle\!\rangle tr_3(\varphi) \text{ for all } a \in Atm$$

4 A Variant of OPDL Interpreted over Labeled Transition Systems

As we have shown in Section 3.2, the logic OPDL interpreted over OPDL models (Definition 3) embeds the variant of CTL* interpreted over $(\mathbb{N} \times W)$ structures which in turn is equivalent to the variant of CTL* interpreted over bundled trees.

A second variant of CTL*, first introduced by [9], is the one interpreted over $\mathcal{R}$-generable models of the form $M = (W, \mathcal{R}, \mathcal{V})$ where W is a set of states, $\mathcal{R} \subseteq W \times W$ is a *total* binary relation on W (*i.e.*, for every $w \in W$, there is some $v \in W$ such that $(w, v) \in \mathcal{R}$) and $\mathcal{V} : W \longrightarrow 2^{Prop}$ is a valuation function for atomic propositions.[3]

Given a model $M = (W, \mathcal{R}, \mathcal{V})$, a *fullpath* in M is defined to be an infinite sequence $(w_1, w_2, w_3, \ldots)$ of states of M such that for each $i \geq 1$, $(w_i, w_{i+1}) \in \mathcal{R}$. Given a fullpath $\sigma = (w_1, w_2, w_3, \ldots)$ and an integer $i \geq 1$, the symbol $\sigma_{\geq i}$ denotes the fullpath $(w_i, w_{i+1}, \ldots)$. As usual, $\sigma[1]$ denotes the first element of the sequence σ. Truth of a CTL* formula is evaluated with respect to a $\mathcal{R}$-generable model M and a fullpath σ in M. Specifically, the rules defining the truth conditions of CTL* formulae are inductively defined as follows:

$$
\begin{aligned}
M, \sigma \models p &\iff p \in \mathcal{V}(\sigma[1]);\\
M, \sigma \models \neg\varphi &\iff M, \sigma \not\models \varphi;\\
M, \sigma \models \varphi_1 \wedge \varphi_2 &\iff M, \sigma \models \varphi_1 \text{ AND } M, \sigma \models \varphi_2;\\
M, \sigma \models \mathsf{X}\varphi &\iff M, \sigma_{\geq 2} \models \varphi\\
M, \sigma \models \varphi\, \mathcal{U}\, \psi &\iff \exists i \geq 1 : M, \sigma_{\geq i} \models \psi \text{ AND } \forall j : \text{IF } 1 \leq j < i \text{ THEN } M, \sigma_{\geq j} \models \varphi\\
M, \sigma \models \mathsf{A}\varphi &\iff \forall \sigma' : \text{IF } \sigma[1] = \sigma'[1] \text{ THEN } M, \sigma' \models \varphi
\end{aligned}
$$

Here we consider a variant of OPDL which embeds the preceding variant of CTL* interpreted over $\mathcal{R}$-generable models. We call OPDL^{lts} this variant of OPDL, where OPDL^{lts} means 'OPDL interpreted over labeled transition systems'. The semantics for OPDL^{lts} is given in terms of PDL models as defined in Section 3.1 (Definition 6), which are nothing but labeled transition systems, *i.e.*, transition systems where transitions between states are labeled with atomic programs. Given a PDL model $M = (W, \{\mathcal{R}_a \mid a \in Atm\}, \mathcal{V})$, let the successor state function $succ$ be defined by $succ(w) = \bigcup_{a \in Atm}\{v \in W \mid (w, v) \in \mathcal{R}_a\}$ for each $w \in W$. $succ(w)$ identifies the successors of world w in M. Moreover, for every $w \in W$, let

$$PA = \{(w_1, \ldots, w_n) \mid w_1, \ldots, w_n \in W \text{ and } w_{i+1} = succ(w_i) \text{ for all } 1 \leq i < n\}$$

be the set of all *paths* in M. For every $w \in W$, let MPA_w be the set of all maximal paths starting in w, also called *histories starting in* w. That is, $\sigma \in MPA_w$ if and only if $\sigma \in PA$ and $\sigma[1] = w$ and there is no $\sigma' \in PA$ such that $\sigma'[1] = w$ and $\sigma \sqsubset \sigma'$ (*i.e.*, σ is a proper initial subsequence of σ'). Finally, let $IN = \{w/\sigma \mid w \in W \text{ and } \sigma \in MPA_w\}$ be the set of all *indexes* in the model M.

[3] It has been proved that the variant of CTL* interpreted over $(\mathbb{N} \times W)$ structures is more general than the variant of CTL* interpreted $\mathcal{R}$-generable models, in the sense that the former have less validities than the latter. For instance, as shown by [22], the formula $\mathsf{AG}(p \rightarrow \mathsf{EX}p) \rightarrow (p \rightarrow \mathsf{EG}p)$ is valid in the latter variant of CTL* but is not valid in the former.

Truth of an OPDL formula is evaluated at a given index $w/\sigma \in IN$ of a PDL model M. The rules inductively defining the truth conditions of OPDL formulae are:

$$\begin{aligned}
M, w/\sigma \models p &\iff p \in \mathcal{V}(w);\\
M, w/\sigma \models \neg\varphi &\iff M, w/\sigma \not\models \varphi;\\
M, w/\sigma \models \varphi \wedge \psi &\iff M, w/\sigma \models \varphi \text{ AND } M, w/\sigma \models \psi;\\
M, w/\sigma \models [\![\pi]\!]\varphi &\iff \forall\, v/\sigma' \in \rho_\pi(w/\sigma) : M, v/\sigma' \models \varphi
\end{aligned}$$

where

$$\begin{aligned}
\rho_a &= \{(w/\sigma, v/\sigma') \mid w/\sigma, v/\sigma' \in IN, (w,v) \in \mathcal{R}_a \text{ and } \sigma = (w, \sigma')\}\\
\rho_{\equiv} &= \{(w/\sigma, v/\sigma') \mid w/\sigma, v/\sigma' \in IN \text{ and } w = v\}\\
\rho_{\pi_1;\pi_2} &= \rho_{\pi_1} \circ \rho_{\pi_2}\\
\rho_{\pi_1 \cup \pi_2} &= \rho_{\pi_1} \cup \rho_{\pi_2}\\
\rho_{\pi^*} &= (\rho_\pi)^*\\
\rho_{\varphi?} &= \{(w/\sigma, w/\sigma) \mid w/\sigma \in IN \text{ and } M, w/\sigma \models \varphi\}
\end{aligned}$$

and $\rho_\pi(w/\sigma) = \{v/\sigma' \mid (w/\sigma, v/\sigma') \in \rho_\pi\}$.

A formula φ of the language $\mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ is said to be OPDL^{lts} valid, denoted by $\models_{\mathsf{OPDL}^{lts}} \varphi$, if and only if φ is true in all PDL models. As usual, a formula φ of the language $\mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ is said to be OPDL^{lts} satisfiable if and only if $\neg\varphi$ is not OPDL^{lts} valid. We shall say that a formula φ of the language $\mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ is a global logical consequence in OPDL^{lts} of a finite set of global axioms $\Gamma = \{\chi_1, \ldots, \chi_n\}$, denoted by $\Gamma \models_{\mathsf{OPDL}^{lts}} \varphi$, if and only if for every PDL model M, if Γ is true in M then φ is true in M too.

As the following theorem shows, the translation given in Section 3.2 provides an embedding of the variant of CTL^* interpreted over $\mathcal{R}$-generable models into OPDL^{lts}.

Theorem 4. *Let φ be a CTL^* formula. φ is valid with respect to the class of $\mathcal{R}$-generable models if and only if $\{\langle\!\langle x \rangle\!\rangle \top\} \models_{\mathsf{OPDL}^{lts}} tr_3(\varphi)$.*

From Theorem 4 and the fact that, as in OPDL, the problem of global logical consequence in OPDL^{lts} with a finite number of global axioms is reducible to the problem of OPDL^{lts} validity, it follows that the satisfiability problem in the variant of CTL^* interpreted over $\mathcal{R}$-generable models can be reduced to the satisfiability problem in OPDL^{lts} with a finite number of atomic programs.

Corollary 2. *Let φ be a CTL^* formula and let Atm be finite. φ is valid with respect to the class of $\mathcal{R}$-generable models if and only if $\models_{\mathsf{OPDL}^{lts}} [\![\mathbf{any}^*]\!]\langle\!\langle x \rangle\!\rangle \top \rightarrow tr_3(\varphi)$.*

5 Perspectives

We have presented a new logic called Ockhamist Propositional Dynamic Logic OPDL and studied its relationship with PDL and CTL^*. An interesting issue for future research is the study of the relationship between OPDL and PDL with intersections of programs. Intersections of *atomic* programs can be simulated in OPDL as follows:

$$[\![a \cap b]\!]\varphi \stackrel{\mathtt{def}}{=} [\![\equiv]\!](\langle\!\langle a \rangle\!\rangle \top \rightarrow [\![b]\!]\varphi)$$

However, it is not clear whether we can find a simple translation from PDL with intersection of (not necessarily atomic) programs to OPDL that preserves validity.

Another direction of future research is the study of the exact complexity of the satisfiability problem for OPDL. The embedding of CTL* into OPDL ensures that it is 2-EXPTIME hard. However, the construction based on the "mosaic method" given in the Section 2.3 does not provide an optimal decision procedure for OPDL. Future work will be devoted to find an optimal decision procedure for OPDL showing that its satisfiability problem is in 2-EXPTIME. Indeed, at the current stage, we conjecture that CTL* is not easier to deal with than OPDL. We also plan to find a sound and complete axiomatization for the logic OPDL.

As to the logic OPDL^{lts} whose semantics has been sketched in Section 4, much work remains to be done. First of all, we plan to study more in detail the differences between OPDL and OPDL^{lts}, taking inspiration from Reynolds' work [22] on the comparison between the CTL* semantics in terms of $\mathcal{R}$-generable models and the CTL* semantics in terms of $(\mathbb{N} \times W)$ structures (or bundled trees). For instance, we plan to find some interesting examples of formulae of the language $\mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ which are valid in OPDL^{lts} but are not valid in OPDL. Secondly, we plan to adapt the proof of the decidability of the satisfiability problem for OPDL given in the Section 2.3 in order to prove the decidability of the satisfiability problem for OPDL^{lts}. Another aspect of the logic OPDL^{lts} that we plan to investigate in the future is its relationship with the extension of PDL with a repetition construct (PDL-Δ) by [26]. PDL-Δ extends PDL with constructions of the form $\Delta\pi$ meaning that "the program π can be repeatedly executed infinitely many times". We believe that the construction $\Delta\pi$ can be simulated in OPDL^{lts} as follows:

$$\Delta\pi \stackrel{\mathtt{def}}{=} \langle\langle\equiv\rangle\rangle[\![\pi^*]\!]\langle\langle\pi\rangle\rangle\top$$

We postpone to future work the definition of the exact translation from PDL-Δ formulae to OPDL^{lts} formulae and a theorem stating that, for every PDL-Δ formula φ, φ is PDL-Δ valid if and only if the translation of φ in OPDL^{lts} is OPDL^{lts} valid.

References

1. Alur, R., Henzinger, T., Kupferman, O.: Alternating-time temporal logic. Journal of the ACM 49, 672–713 (2002)
2. Axelsson, R., Hague, M., Kreutzer, S., Lange, M., Latte, M.: Extended computation tree logic. In: Fermüller, C.G., Voronkov, A. (eds.) LPAR-17. LNCS, vol. 6397, pp. 67–81. Springer, Heidelberg (2010)
3. Balbiani, P., Lorini, E.: Ockhamist propositional dynamic logic: a natural link between PDL and CTL*. Technical Report IRIT/RT–2013-12–FR, Institut de Recherche en Informatique de Toulouse (2013)
4. Belnap, N., Perloff, M., Xu, M.: Facing the future: agents and choices in our indeterminist world. Oxford University Press (2001)
5. Brown, M., Goranko, V.: An extended branching-time ockhamist temporal logic. Journal of Logic, Language and Information 8(2), 143–166 (1999)
6. Clarke, E.M., Emerson, E.A.: Design and synthesis of synchronization skeletons using branching time temporal logic. In: Kozen, D. (ed.) Logic of Programs 1981. LNCS, vol. 131, pp. 52–71. Springer, Heidelberg (1982)

7. Dam, M.: CTL* and ECTL* as fragments of the modal mu-calculus. Theoretical Computer Science 126(1), 77–96 (1994)
8. De Nicola, R., Vaandrager, F.W.: Action versus state based logics for transition systems. In: Guessarian, I. (ed.) LITP 1990. LNCS, vol. 469, pp. 407–419. Springer, Heidelberg (1990)
9. Emerson, E.A., Halpern, J.: 'Sometimes' and 'not never' revisited: on branching versus linear time. Journal of the ACM 33, 151–178 (1986)
10. Emerson, E.A., Sistla, A.: Deciding full branching time logic. Information and Control 61, 175–201 (1984)
11. Emerson, E.A.: Temporal and modal logic. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science. Formal Models and Semantics, vol. B. North-Holland Pub. Co./MIT Press (1990)
12. Fischer, M.J., Ladner, R.E.: Propositional dynamic logic of regular programs. Journal of Computer Systems Science 18(2), 194–211 (1979)
13. Goranko, V.: Coalition games and alternating temporal logics. In: Proc. of TARK 2001, pp. 259–272. Morgan Kaufmann (2001)
14. Harel, D.: Dynamic logic. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, vol. 2, pp. 497–604. Reidel, Dordrecht (1984)
15. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press (2000)
16. Masini, A., Viganò, L., Volpe, M.: Labelled natural deduction for a bundled branching temporal logic. Journal of Logic and Computation 21(6), 1093–1163 (2011)
17. McCabe-Dansted, J.C.: A tableau for the combination of CTL and BCTL. In: Proc. of TIME 2012, pp. 29–36. IEEE Computer Society (2012)
18. Németi, I.: Decidable versions of first order logic and cylindric-relativized set algebras. In: Csirmaz, L., Gabbay, D., de Rijke, M. (eds.) Logic Colloquium 1992, pp. 171–241. CSLI Publications (1995)
19. Nishimura, H.: Descriptively complete process logic. Acta Informatica 14, 359–369 (1980)
20. Pnueli, A.: The temporal logic of programs. In: Proc. of the Eighteenth Symposium on Foundations of Computer Science, pp. 46–57. IEEE Computer Society (1977)
21. Prior, A.: Past, Present, and Future. Clarendon Press, Oxford (1967)
22. Reynolds, M.: An axiomatization of full computation tree logic. Journal of Symbolic Logic 66(3), 1011–1057 (2001)
23. Reynolds, M.: A tableau for bundled CTL*. Journal of Logic and Computation 17(1), 117–132 (2007)
24. Reynolds, M.: A tableau for CTL*. In: Cavalcanti, A., Dams, D.R. (eds.) FM 2009. LNCS, vol. 5850, pp. 403–418. Springer, Heidelberg (2009)
25. Stirling, C.: Modal and temporal logics. In: Abramsky, S., Gabbay, D.M., Maibaum, T.S.E. (eds.) Handbook of Logic in Computer Science, vol. 2. Clarendon Press, Oxford (1992)
26. Streett, R.S.: Propositional dynamic logic of looping and converse is elementarily decidable. Information and Control 54(1-2), 121–141 (1982)
27. Thomason, R.: Combinations of tense and modality. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, vol. 2, pp. 135–165. Reidel, Dordrecht (1984)
28. Vardi, M.Y., Stockmeyer, L.J.: Improved upper and lower bounds for modal logics of programs. In: Proc. of the 17th Annual ACM Symposium on Theory of Computing, pp. 240–251. ACM (1985)
29. Wolper, P.: A translation from full branching time temporal logic to one letter propositional dynamic logic with looping. Unpublished manuscript (1982)
30. Zanardo, A.: Branching-time logic with quantification over branches: The point of view of modal logic. Journal of Symbolic Logic 61(1), 143–166 (1996)