Dynamic description logics

Frank Wolter and Michael Zakharyaschev*
Institut für Informatik, Universität Leipzig
Augustus-Platz 10-11, 04109 Leipzig, Germany;
Keldysh Institute for Applied Mathematics
Russian Academy of Sciences
Miusskaya Square 4, 125047 Moscow, Russia
(e-mails: wolter@informatik.uni-leipzig.de, mz@spp.keldysh.ru)

Section: *algebraic and model-theoretic aspects of modal logic*

*The work of the second author was supported by the Russian Fundamental Research Foundation.

1 Introduction

The topic of this paper can be viewed from different standpoints. A modal logician would probably say that we combine polymodal **K** with **PDL** and prove the decidability of the resulting hybrid. In the field of knowledge representation, the paper can be characterized as an attempt to introduce a dynamic dimension in concept description (alias terminological) logics. And finally, in a broader perspective, our concern is to construct and study formalisms for representing and processing knowledge in dynamic application domains that would be maximally expressive, on the one hand, and decidable, on the other.

Concept description (or simply description) logics originate from practical knowledge representation systems (see e.g. [3, 8, 1]) which, in turn, can be traced back to the ideas of semantic networks and frames. An application domain is represented in the framework of a description logic by means of formulas which define complex concepts out of primitive ones and assert that certain objects belong to certain concepts or are in certain relations to some other objects. Starting, for instance, from the primitive concepts child, $\mathsf{grandma}$, $\mathsf{wealthy}$, $\mathsf{warm\_island}$ and the binary relations (or roles) has, lives we can define a compound concept

$$\mathsf{fortunate\_child} = \mathsf{child} \wedge \exists \mathsf{has}.(\mathsf{grandma} \wedge \mathsf{wealthy} \wedge \exists \mathsf{lives}.\mathsf{warm\_island})$$

comprising all children whose grandmothers are wealthy and live on warm islands. The formulas $John : \mathsf{fortunate\_child}$, $Mary\ \mathsf{lives}\ Bahamas$ assert that John is a fortunate child and that Mary lives on Bahamas.

The relativized existential quantifier $\exists R$ has the same semantic meaning as the possibility operator $\Diamond$ interpreted by the accessibility relation R. This observation, first made by Schild [13], establishes a close connection between description logics and polymodal **K**. Many other concept and role constructs used in description logics have their modal counterparts as well, for instance, number restrictions, nominals, and transitive reflexive closures of roles.

Description logics were originally designed for representing only static knowledge. To take into account changes in time or under certain actions and retain the relative simplicity of the language (say, decidability) it is natural to extend it by the corresponding modal operators and thereby keep its propositional modal status. It is known, however, that combinations of rather simple modal systems may result in very complex ones (see e.g. [18]). The first temporal and epistemic description logics constructed in [16, 14, 9] were either too expressive and consequently undecidable or too weak (the temporal operators were applicable either only to formulas or only to concepts). A compromise was found by Baader and Laux [2] who combined the description logic $\mathcal{ALC}$ of [15] with polymodal **K** by allowing applications of modal operators to both formulas and concepts and showed the decidability of the satisfiability problem for the resulting language in models with expanding domains.

In [19, 20] we have launched a systematic investigation of description logics with modal operators and proved the decidability of various epistemic and temporal description logics under the constant domain assumption. In this paper we combine description logics with propositional dynamic logic **PDL**.

PDL was originally conceived (see [11, 7]) as a formalism for reasoning about the behaviour of non-deterministic iterative programs described by regular expressions over a set of atomic programs and tests. It turned out to be also useful as a basis for a logic of action and planning in artificial intelligence [17, 12] and for deontic logic [10]. Many other types of modal logics can be regarded as fragments of **PDL**, for instance, polymodal **K**, **S4**, various epistemic logics with the common knowledge operator.

We increase the expressive power of **PDL** by extending its propositional basis to the language of description logics. The modal operators of **PDL** can be applied to both concepts and formulas. For example, we can define concepts like

$$\mathsf{easy\_cured\_child} = \mathsf{child} \wedge \exists \mathsf{has.angina} \wedge \langle (\mathsf{give\_honey} \cup \mathsf{give\_aspirin})^* \rangle \neg \exists \mathsf{has.angina}$$

(i.e., the set of children suffering from angina that can be cured by using honey and aspirin).

Our aim is to develop a technique of proving the decidability of the satisfiability problem for the resulting logics (which do not in general enjoy the finite model property). For simplicity we will be dealing here only with $\mathcal{ALC}$; however it can be replaced with more expressive description logics provided that they are decidable.

2 Syntax and semantics of $\mathcal{PDLC}$

We begin by defining the dynamic concept description language $\mathcal{PDLC}$ and its semantics.

Definition 1 (alphabet). The *primitive symbols* of $\mathcal{PDLC}$ are:

- *concept names* $C_0, C_1, \dots$;
- *role names* $R_0, R_1, \dots$;
- *object names* $a_0, a_1, \dots$;
- the booleans (say, $\wedge$, $\neg$, $\top$) and the *relativized existential quantifier* $\exists R_i$, for every role name R_i;
- *action variables* $\alpha_0, \alpha_1, \dots$;
- *action constructs*: ; (composition), $\cup$ (alternation), * (iteration), ? (test).

Now we define by induction the notions of a concept, a formula and an action term. Concepts will usually be denoted by symbols C and D, formulas by φ, ψ and χ, while for action terms we reserve small Greek characters from the beginning of the alphabet, α, β, etc.

Definition 2 (concept, formula, action term). Every concept name as well as $\top$ is an *atomic concept*. Every action variable is an *atomic action*. If C and D are concepts, a and b object names, R a role name, φ and ψ formulas, α and β action terms, then

- $C \wedge D$, $\neg C$, $\exists R.C$, $[\alpha]C$ are *concepts*;
- $a : C$, aRb, $\varphi \wedge \psi$, $\neg\varphi$, $[\alpha]\varphi$ are *formulas* (the first two of them are *atomic*);
- $\alpha;\beta$, $\alpha \cup \beta$, α^*, $\varphi?$ are *action terms*.

The pure description part of the language $\mathcal{PDLC}$ is the standard concept description language $\mathcal{ALC}$ (see [6]); it is interpreted in models of the form

$$I = \left\langle \Delta, R_0^I, \dots, C_0^I, \dots, a_0^I, \dots \right\rangle,$$

where Δ is a set of objects, R_i^I is a binary relation on Δ interpreting the role name R_i, C_i^I is a subset of Δ interpreting the concept name C_i, and $a_i^I \in \Delta$ interprets the object name a_i.

The dynamic component of $\mathcal{PDLC}$ is the language of the well known propositional dynamic logic **PDL** (see e.g. [7]). It is interpreted in frames (or labelled transition systems) of the form

$$\mathfrak{T} = \langle W, T_{\alpha_0}, T_{\alpha_1} \dots \rangle, \tag{1}$$

where W is a non-empty set of *states* and T_{α_i} a binary relation on W interpreting transitions corresponding to the action variable α_i.

By combining these two kinds of models we arrive at the following definition.

Definition 3 (model). A *model* of $\mathcal{PDLC}$ *based on a frame* $\mathfrak{T}$ of the form (1) is a pair $\mathfrak{M} = \langle \mathfrak{T}, I \rangle$ in which I is a function associating with each state $w \in W$ an $\mathcal{ALC}$-model

$$I(w) = \left\langle \Delta, R_0^{I(w)}, \dots, C_0^{I(w)}, \dots, a_0^{I(w)}, \dots \right\rangle$$

such that $a_i^{I(u)} = a_i^{I(v)}$ for any $u, v \in W$. Note that the set of objects Δ is the same for every state in W[1]; it is called the *domain* of $\mathfrak{M}$.

It is also worth emphasizing that the interpretation of the object names does not depend on the particular world, which means that we use *rigid designators*. With minor changes we could also take into account the unique object name assumption according to which $a_i^{I(w)} \neq a_j^{I(w)}$ whenever $i \neq j$.

Definition 4 (satisfaction). Given a $\mathcal{PDLC}$-model $\mathfrak{M} = \langle \mathfrak{T}, I \rangle$ and a state w in it, the *value* $C^{I(w)}$ of a concept C in w, the *truth-relation* $(\mathfrak{M}, w) \models \varphi$ (or simply $w \models \varphi$ if $\mathfrak{M}$ is understood), and the relation T_α, α an action term, are defined inductively as follows:

1. $\top^{I(w)} = \Delta$ and $C^{I(w)} = C_i^{I(w)}$, for $C = C_i$;
2. $(C \wedge D)^{I(w)} = C^{I(w)} \cap D^{I(w)}$;
3. $(\neg C)^{I(w)} = \Delta - C^{I(w)}$;
4. $x \in (\exists R_i.C)^{I(w)}$ iff $\exists y \in C^{I(w)}\ x R_i^{I(w)} y$;
5. $x \in ([\alpha]C)^{I(w)}$ iff $\forall v \in W\ (w T_\alpha v \Rightarrow x \in C^{I(v)})$;
6. $w \models C = D$ iff $C^{I(w)} = D^{I(w)}$;
7. $w \models a : C$ iff $a^{I(w)} \in C^{I(w)}$;
8. $w \models aRb$ iff $a^{I(w)} R^{I(w)} b^{I(w)}$;
9. $w \models \varphi \wedge \psi$ iff $w \models \varphi$ and $w \models \psi$;
10. $w \models \neg\varphi$ iff $w \not\models \varphi$;
11. $w \models [\alpha]\varphi$ iff $\forall v \in W\ (w T_\alpha v \Rightarrow v \models \varphi)$;
12. $T_{\varphi?} = \{\langle w, w \rangle : w \models \varphi\}$;
13. $T_{\alpha;\beta} = T_\alpha \circ T_\beta$ (the composition of T_α and T_β);
14. $T_{\alpha \cup \beta} = T_\alpha \cup T_\beta$;
15. $T_{\alpha^*} = T_\alpha^*$ (the transitive and reflexive closure of T_α).

A formula φ is *satisfiable* if there are a $\mathcal{PDLC}$-model $\mathfrak{M}$ and a state w in $\mathfrak{M}$ such that $w \models \varphi$.

The main goal of this paper is to develop a satisfiability checking algorithm for $\mathcal{PDLC}$-formulas. It is to be noted that although both $\mathcal{ALC}$ and **PDL** have the finite model property, their hybrid defined above does not enjoy it: as follows from [19], there is a $\mathcal{PDLC}$-formula satisfiable in an infinite model but not in finite ones. The satisfiability checking algorithm we are going to construct in Section 4 is based on the mosaic technique and the representation of models in the form of quasimodels.

[1] This means that we accept the *constant domain assumption*.

3 Quasimodels

The aim of this section is to show that (modulo a given formula) every $\mathcal{PDLC}$-model can be represented as a structure, called a quasimodel, every state in which is finite. Let us fix a $\mathcal{PDLC}$-formula φ and denote by $sub\varphi$, $con\varphi$ and $ob\varphi$ the sets of all subformulas, concepts and object names occurring in φ, respectively.

Definition 5 (Fischer–Ladner closure). The *Fischer–Ladner closure* of φ is the pair $\langle\Phi(\varphi), \Lambda(\varphi)\rangle$ in which $\Phi(\varphi) \supseteq sub\varphi$ and $\Lambda(\varphi) \supseteq con\varphi$ are the smallest sets of formulas and concepts that are closed under subformulas and subconcepts, respectively, and satisfy the following conditions:

- $[\alpha;\beta]\psi \in \Phi(\varphi) \Rightarrow [\alpha][\beta]\psi \in \Phi(\varphi)$;
- $[\alpha \cup \beta]\psi \in \Phi(\varphi) \Rightarrow [\alpha]\psi, [\beta]\psi \in \Phi(\varphi)$;
- $[\alpha^*]\psi \in \Phi(\varphi) \Rightarrow [\alpha][\alpha^*]\psi \in \Phi(\varphi)$;
- $[\psi?]\chi \in \Phi(\varphi) \Rightarrow \psi \in \Phi(\varphi)$;
- $[\alpha;\beta]C \in \Lambda(\varphi) \Rightarrow [\alpha][\beta]C \in \Lambda(\varphi)$;
- $[\alpha \cup \beta]C \in \Lambda(\varphi) \Rightarrow [\alpha]C, [\beta]C \in \Lambda(\varphi)$;
- $[\alpha^*]C \in \Lambda(\varphi) \Rightarrow [\alpha][\alpha^*]C \in \Lambda(\varphi)$;
- $[\psi?]C \in \Lambda(\varphi) \Rightarrow \psi \in \Phi(\varphi)$.

Definition 6 (quasistate). Consider a structure of the form

$$q = \langle X_q, R_0^q, \ldots, C_0^q, \ldots, ([\beta_0]D_0)^q, \ldots, a_0^q, \ldots, \Xi^q\rangle. \qquad (2)$$

Here X_q is a finite set, the *domain* of q, $R_i^q \subseteq X_q \times X_q$ for every role name R_i in φ, $C_i^q \subseteq X_q$ for every $C_i \in con\varphi$, $a_i^q \in X_q$ for every $a_i \in ob\varphi$, $([\beta_i]D_j)^q \subseteq X_q$ for every $[\beta_i]D_j$ in $\Lambda(\varphi)$, and Ξ^q is a subset of $\Phi(\varphi)$. The *value* C^q of a concept $C \in \Lambda(\varphi)$ in q is computed almost in the same way as in Definition 4, the only difference is that now the value $([\alpha]C)^q$ is given in q directly as a value of an atomic concept. We call q a *quasistate* for φ if the following conditions hold:

- $([\alpha;\beta]C)^q = ([\alpha][\beta]C)^q$;
- $([\alpha \cup \beta]C)^q = ([\alpha]C)^q \cap ([\beta]C)^q$;
- $([\alpha^*]C)^q = C^q \cap ([\alpha][\alpha^*]C)^q$;
- $([\psi?]C)^q = \{x \in X_q : \psi \in \Xi^q \Rightarrow x \in C^q\}$;
- $C = D \in \Xi^q$ iff $C^q = D^q$, for every $C = D \in \Phi(\varphi)$;
- $a : C \in \Xi^q$ iff $a^q \in C^q$, for every $a : C \in \Phi(\varphi)$;
- $aRb \in \Xi^q$ iff $a^q R^q b^q$, for every $aRb \in \Phi(\varphi)$;
- $\psi \wedge \chi \in \Xi^q$ iff $\psi \in \Xi^q$ and $\chi \in \Xi^q$, for every $\psi \wedge \chi \in \Phi(\varphi)$;
- $\neg\psi \in \Xi^q$ iff $\psi \notin \Xi^q$, for every $\neg\psi \in \Phi(\varphi)$;
- $[\alpha;\beta]\psi \in \Xi^q$ iff $[\alpha][\beta]\psi \in \Xi^q$, for every $[\alpha;\beta]\psi \in \Phi(\varphi)$;
- $[\alpha \cup \beta]\psi \in \Xi^q$ iff $[\alpha]\psi, [\beta]\psi \in \Xi^q$, for every $[\alpha \cup \beta]\psi \in \Phi(\varphi)$;
- $[\alpha^*]\psi \in \Xi^q$ iff $\psi, [\alpha][\alpha^*]\psi \in \Xi^q$, for every $[\alpha^*]\psi \in \Phi(\varphi)$;

- $[\psi?]\chi \in \Xi^q$ iff $\psi \notin \Xi^q$ or $\chi \in \Xi^q$, for every $[\psi?]\chi \in \Phi(\varphi)$.

Instead of $\psi \in \Xi^q$ we will often write $q \models \psi$ and say that ψ is *true* in q.

Given a structure of the form (2), we can always effectively decide whether it is a quasistate for φ.

Let $\mathfrak{m} = \langle Q, T_{\alpha_1}, \dots, T_{\alpha_k}\rangle$ be a frame in which Q is a set of quasistates for φ and T_{α_i} is a binary relation on Q for every action variable α_i in φ. For an action term α constructed from the action variables $\alpha_1, \dots, \alpha_k$, let T_α be the binary relation on Q defined by items 12–15 of Definition 4 (in which w should be replaced by q).

Definition 7 (run). By a *run* in $\mathfrak{m} = \langle Q, T_{\alpha_1}, \dots, T_{\alpha_k}\rangle$ we will mean a set r which contains precisely one object from the domain X_q of each quasistate $q \in Q$ — let us denote this object by $r(q)$ — and such that for every concept $[\alpha]C \in \Lambda(\varphi)$ and every $q \in Q$ we have:

- $r(q) \in ([\alpha]C)^q \Leftrightarrow \forall q' \in Q\ (qT_\alpha q' \Rightarrow r(q') \in C^{q'})$.

If only the $(\Rightarrow)$-part of this condition holds then r is called a *weak run*.

Definition 8 (quasimodel). The frame $\mathfrak{m} = \langle Q, T_{\alpha_1}, \dots, T_{\alpha_k}\rangle$ is a *quasimodel* for φ if

- for every $q \in Q$ and every $x \in X_q$ there is a run r in $\mathfrak{m}$ such that $r(q) = x$;
- for every $a \in ob\varphi$, $r_a = \{a^q : q \in Q\}$ is a run in $\mathfrak{m}$;
- for every $[\alpha]\psi \in \Phi(\varphi)$ and every $q \in Q$,

$$q \models [\alpha]\psi \Leftrightarrow \forall q' \in Q\ (qT_\alpha q' \Rightarrow q' \models \psi). \qquad (3)$$

A formula φ is *satisfiable* in $\mathfrak{m}$ if $q \models \varphi$ for some $q \in Q$. If in this definition we replace runs with weak runs and require that only the $(\Rightarrow)$-part of condition (3) holds then $\mathfrak{m}$ will be called a *weak quasimodel* for φ.

Theorem 9 (quasimodel completeness). *A formula φ is satisfiable iff it is satisfiable in some quasimodel for φ.*

Proof $(\Rightarrow)$ Suppose φ is satisfiable in a model $\mathfrak{M} = \langle \mathfrak{T}, I\rangle$ based on a frame $\mathfrak{T}$ of the form (1) and having a domain Δ. With each $w \in W$ we associate a quasistate

$$q_w = \langle X_{q_w}, R_0^{q_w}, \dots, C_0^{q_w}, \dots, ([\beta_0]D_0)^{q_w}, \dots, a_0^{q_w}, \dots, \Xi^{q_w}\rangle$$

in the following way. For every $x \in \Delta$ let

$$t_w(x) = \{C \in \Lambda(\varphi) : x \in C^{I(w)}\}.$$

Then X_{q_w} contains the objects $a \in ob\varphi$ from Δ (without loss of generality we may assume $a^{I(w)} = a$) and also one representative $z \notin ob\varphi$ from each class $[x]_w = \{y \in \Delta : t_w(x) = t_w(y)\}$, if such z exists, $xR_i^{q_w}y$ iff either one of x, y is not in $ob\varphi$ and $x'R_i^{I(w)}y'$ for some $x' \in [x]_w$, $y' \in [y]_w$, or $x, y \in ob\varphi$ and $xR_i^{I(w)}y$, $x \in C_i^{q_w}$ iff $x \in C_i^{I(w)}$, $x \in ([\beta_i]D_j)^{q_w}$ iff $x \in ([\beta_i]D_j)^{I(w)}$, and $\Xi^{q_w} = \{\psi \in \Phi(\varphi) : w \models \psi\}$. It is a matter of routine to check by induction that for every $C \in \Lambda(\varphi)$, we have $C^{q_w} = C^{I(w)} \cap X_{q_w}$ and so q_w is a quasistate for φ. The structure

$$\mathfrak{m} = \langle Q, T'_{\alpha_1}, \dots, T'_{\alpha_n}\rangle,$$

where $Q = \{q_w : w \in W\}$ and $q_u T'_{\alpha_i} q_v$ iff $uT_{\alpha_i}v$, for every action variable α_i in φ and all $u, v \in W$, is then a quasimodel satisfying φ (to construct a run through a given $x \in X_{q_u}$, one can take any $r(q_w) \in [x]_w \cap X_{q_w}$, for every $w \in W$).

($\Leftarrow$) Now let $\mathfrak{m} = \langle Q, T_{\alpha_1}, \ldots, T_{\alpha_n} \rangle$ be a quasimodel for φ such that $q \models \varphi$ for some $q \in Q$. Construct a standard model $\mathfrak{M} = \langle \mathfrak{m}, I \rangle$ based on the frame $\mathfrak{m}$ by taking, for every $q \in Q$,

$$I(q) = \langle \Delta, R_0^{I(q)}, \ldots, C_0^{I(q)}, \ldots, r_{a_0}, \ldots \rangle,$$

where Δ is the set of all runs in $\mathfrak{m}$, $rR_i^{I(q)}r'$ iff $r(q)R_i^q r'(q)$, and $r \in C_i^{I(q)}$ iff $r(q) \in C_i^q$. By a straightforward induction one can show that for all $C \in \Lambda(\varphi)$, $\psi \in \Phi(\varphi)$, $q \in Q$ and $r \in \Delta$, we have $r \in C^{I(q)}$ iff $r(q) \in C^q$, and $(\mathfrak{M}, q) \models \psi$ iff $\psi \in \Xi^q$. Therefore, φ is satisfied in $\mathfrak{M}$. ❑

It is worth noting that as a consequence of the proof of ($\Rightarrow$) we obtain

Corollary 10. *A formula φ is satisfiable iff it is satisfiable in a quasimodel containing at most*

$$\sharp(\varphi) = 2^{2^{|\Lambda(\varphi)|}} \cdot 2^{|\Phi(\varphi)|} \cdot |ob\varphi| \cdot 2^{|\Lambda(\varphi)|}$$

pairwise non-isomorphic quasistates the cardinality of the domains in which does not exceed

$$\flat(\varphi) = 2^{|\Lambda(\varphi)|} + |ob\varphi|.$$

As is well known, **PDL** is complete with respect to models based on intransitive trees. We remind the reader that a frame $\langle W, R \rangle$ is an intransitive tree if it is rooted, cycle-free, and contains no distinct paths of the form $xRy_1R \ldots Ry_nRy$ and $xRz_1R \ldots Rz_mRy$.

Definition 11 (tree quasimodel). A (weak) quasimodel

$$\mathfrak{m} = \langle Q, T_{\alpha_1}, \ldots, T_{\alpha_k} \rangle$$

for φ is called a *tree (weak) quasimodel* if $T_{\mathfrak{m}} = \bigcup\{T_{\alpha_i} : 1 \leq i \leq k\}$ is an intransitive tree order on Q and $T_{\alpha_i} \cap T_{\alpha_j} = \emptyset$ whenever $i \neq j$. If no quasistate of a tree (weak) quasimodel $\mathfrak{m}$ different from its root has more than one $T_{\mathfrak{m}}$-successor then $\mathfrak{m}$ is called a *bouquet (weak) quasimodel*.

Theorem 12 (tree quasimodel completeness). *A formula φ is satisfiable iff it is satisfiable in a tree quasimodel for φ the domains of quasistates in which are of cardinality $\leq \flat(\varphi)$.*

4 Effective satisfiability criterion

Assume again that we have fixed a $\mathcal{PDLC}$-formula φ.

Definition 13 (block). Let $\mathfrak{b} = \langle Q, T_{\alpha_1}, \ldots, T_{\alpha_k} \rangle$ be a finite bouquet weak quasimodel with root q_0. Say that a weak run r in $\mathfrak{b}$ is *root-saturated* if

$$\forall [\alpha]C \in \Lambda(\varphi)\ (r(q_0) \notin ([\alpha]C)_0^q \Rightarrow \exists q' \in Q\ (q_0 T_\alpha q'\ \&\ r(q') \notin C^{q'})).$$

We call $\mathfrak{b}$ a *block* for φ if

(a) for every $q \in Q$ and every $x \in X_q$ there is a root-saturated weak run r in $\mathfrak{b}$ such that $r(q) = x$;

(b) every weak run r_a, $a \in ob\varphi$, is root-saturated;

(c) $\forall [\alpha]\psi \in \Phi(\varphi)\ (q_0 \not\models [\alpha]\psi \Rightarrow \exists q' \in Q\ (q_0 T_\alpha q'\ \&\ q' \not\models \psi))$.

Definition 14 (satisfying set). A set $\mathcal{S}$ of blocks for φ is called a *satisfying set* for φ if (i) it contains a block with root q_0 such that $q_0 \models \varphi$ and (ii) for every quasistate q in every block in $\mathcal{S}$ there exists a block in $\mathcal{S}$ having q as its root.

Our aim is to show that φ is satisfiable iff there is a satisfying set for φ whose blocks contain at most N quasistates, for some $N < \omega$ effectively determined by φ.

Denote by $|\alpha|$ the *length* of an action term α which is defined inductively as follows

- $|\alpha_i| = 1$, $|\psi?| = 1$;
- $|\alpha \cup \beta| = \max\{|\alpha|, |\beta|\}$;
- $|\alpha; \beta| = |\alpha| + |\beta|$;
- $|\alpha^*| = |\alpha| + 1$.

Now for every $n \geq 0$ we put

- $\alpha_i(n) = \alpha_i$, $\psi?(n) = \psi?$;
- $(\alpha \cup \beta)(n) = \alpha(n) \cup \beta(n)$;
- $(\alpha; \beta)(n) = \alpha(n); \beta(n)$;
- $\alpha^*(n) = \alpha^{\leq n}(n)$,

where

$$\alpha^{\leq n} = \top? \cup \alpha \cup (\alpha; \alpha) \cup \cdots \cup (\underbrace{\alpha; \ldots; \alpha}_{n}).$$

In other words, $\alpha(n)$ results from α by replacing every occurrence of an action term of the form γ^* (which is not in the scope of a test $\psi?$) with $\gamma^{\leq n}$. In particular, $\alpha(n)$ contains no occurrence of *. Finally, let

$$l(\varphi) = \max\{|\alpha(\flat(\varphi) \cdot \sharp(\varphi))| : [\alpha]C \in \Lambda(\varphi) \text{ or } [\alpha]\psi \in \Phi(\varphi)\}.$$

We are in a position now to prove the main result of the paper.

Theorem 15 (satisfiability criterion). *A $\mathcal{PDLC}$-formula φ is satisfiable iff there is a satisfying set for φ each block in which contains at most*

$$N = l(\varphi) \cdot (|\Phi(\varphi)| + 2\flat(\varphi) \cdot |\Lambda(\varphi)|)$$

quasistates whose domains are of cardinality $\leq \flat(\varphi)$.

Proof $(\Rightarrow)$ Suppose φ is satisfiable. Then, by Theorem 12, there is a tree quasimodel $\mathfrak{m} = \langle Q, T_{\alpha_1}, \ldots, T_{\alpha_k} \rangle$ satisfying φ at its root and having quasistates of size $\leq \flat(\varphi)$.

We begin our construction of a satisfying set for φ by associating with each quasistate q in $\mathfrak{m}$ a block $\mathfrak{b}_q = \langle Q^q, T^q_{\alpha_1}, \ldots, T^q_{\alpha_k} \rangle$. First, for every formula $[\alpha]\psi \in \Phi(\varphi)$ such that $q \not\models [\alpha]\psi$ we select a quasistate $q' \in Q$ for which $qT_\alpha q'$, $q' \not\models \psi$, and put it in $Sel(q)$ (at the very beginning $Sel(q) = \emptyset$). Then, for every $x \in X_q$ we fix a run r in $\mathfrak{m}$ coming through x (if $x = a$, for $a \in ob\varphi$, then $r = r_a$) and for every concept $[\alpha]C \in \Lambda(\varphi)$ such that $r(q) = x \notin ([\alpha]C)^q$, select a quasistate $q' \in Q$ for which $qT_\alpha q'$, $r(q') \notin C^{q'}$ and put it in $Sel(q)$ *together with its copy q''*. (Formally, taking the copy q'' means that we duplicate the subtree of $\mathfrak{m}$ generated by q' and connect it with the immediate predecessor $q^\dagger$ of q' by the same relation T_{α_i} that connects $q^\dagger$ with q'. The resulting structure is clearly again a tree quasimodel satisfying φ.)

The number of selected quasistates does not exceed $|\Phi(\varphi)|+2\flat(\varphi)\cdot|\Lambda(\varphi)|$; without loss of generality we may assume these quasistates to be pairwise distinct (otherwise we can duplicate them as above). For each selected q' there is a unique $T_{\mathfrak{m}}$-path from q to q', namely $(q,q')=\{q_1 : qT^*_{\mathfrak{m}}q_1T^*_{\mathfrak{m}}q'\}$. Again, without loss of generality we assume that distinct paths (q,q') and (q,q'') (for $q'\neq q''$) have no common quasistates save q (otherwise the duplication technique will do the job).

Finally, we define Q^q to be the set of all quasistates in the paths (q,q'), for $q'\in Sel(q)$, and $T^q_{\alpha_i}$ to be the restriction of T_{α_i} to Q^q (taking into account all the duplications, of course).

The constructed structure $\mathfrak{b}_q$ is a block. Indeed, it is clearly a finite bouquet weak quasimodel for φ with root q (for it may be considered as a subquasimodel of $\mathfrak{m}$) satisfying (b) and (c) by the construction. Suppose now that $q_1\in Q^q$, $x\in X_{q_1}$, and let r be a weak run in $\mathfrak{b}_q$ coming through x. Consider the object $r(q)$ and the set $\mathcal{C}$ of concepts $[\alpha]C\in\Lambda(\varphi)$ for which $r(q)\notin([\alpha]C)^q$. For each of them there is a weak run $r_{[\alpha]C}$ such that $r_{[\alpha]C}(q)=r(q)$, $r_{[\alpha]C}(q_{[\alpha]C})\notin C^{q_{[\alpha]C}}$, for some $q_{[\alpha]C}\in Sel(q)$, and $q_1\notin(q,q_{[\alpha]C})$. Using these weak runs and r we construct a set r' by taking, for every $q'\in Q^q$,

$$r'(q')=\begin{cases} r(q') & \text{if } q'\notin(q,q_{[\alpha]C}), \text{ for any } [\alpha]C\in\mathcal{C}\\ r_{[\alpha]C}(q') & \text{otherwise}\end{cases}$$

Clearly, r' is a root-saturated weak run in $\mathfrak{b}_q$ coming through x, which establishes (a).

Although the number of branches in $\mathfrak{b}_q$ does not exceed $|\Phi(\varphi)|+2\flat(\varphi)\cdot|\Lambda(\varphi)|$, they may be too long. Our next step is to extract from $\mathfrak{b}_q$ a substructure $\mathfrak{a}_q$ which is still a block for φ and whose branches are of length $\leq l(\varphi)$. We will do this by cutting out certain fragments of branches in $\mathfrak{b}_q$.

Consider a branch (q,q') and suppose that q' was selected to "saturate" $[\alpha]\psi$ in q or $[\alpha]C$ in some $x\in X_q$. If the action term α contains no occurrence of * then, since $qT_\alpha q'$, the length of (q,q') is at most $|\alpha|$; in this case we leave this branch as it is.

Suppose now that α contains iteration. The "truncation" is conducted by induction on the construction of α. If $\alpha=\beta;\gamma$ then $qT^q_\beta q^\dagger T^q_\gamma q'$, for some $q^\dagger\in(q,q')$, and we proceed by truncating $(q,q^\dagger)$ and $(q^\dagger,q')$. If $\alpha=\beta\cup\gamma$ then either $qT^q_\beta q'$ or $qT^q_\gamma q'$ which reduces the complexity of α. Finally, let $\alpha=\beta^*$. Then there are quasistates $q_1,\dots,q_n\in(q,q')$, $n<\omega$, such that $q=q_1T^q_\beta q_2T^q_\beta\dots T^q_\beta q_n=q'$. If $n\leq\sharp(\varphi)\cdot\flat(\varphi)$ then we proceed by considering the fragments (q_i,q_{i+1}) for $1\leq i<n$. Otherwise let r be the weak run in $\mathfrak{b}_q$ such that $r(q)=x$ and $r(q')\notin C^{q'}$; if q' "saturates" $[\alpha]\psi$ in q then r may be any weak run in $\mathfrak{b}_q$. Since $n>\sharp(\varphi)\cdot\flat(\varphi)$, there must be two isomorphic quasistates q_i and q_j, $1\leq i<j\leq n$, such that $t(r(q_i))=t(r(q_j))$. Then we cut out from (q,q') all the quasistates in the interval (q_i,q_j) save q_i and put $q_iT^q_\gamma q^\dagger$ iff $q_jT^q_\gamma q^\dagger$, for every action variable γ. It should be clear that the resulting structure is still a block for φ, and so by deleting repeating quasistates in the branches of $\mathfrak{b}_q$ we can construct a block $\mathfrak{a}_q$ for φ whose branches are of length $\leq l(\varphi)$.

The satisfying set for φ we are looking for can be constructed now by taking the blocks $\mathfrak{a}_q$ for all non-isomorphic quasistates in $\mathfrak{m}$.

($\Leftarrow$) Let $\mathcal{S}$ be a satisfying set for φ. We are going to construct a quasimodel $\mathfrak{m}$ satisfying φ as the limit of a sequence of weak quasimodels

$$\mathfrak{m}_n=\langle Q_n,R^n_{\alpha_1},\dots,R^n_{\alpha_k}\rangle,\ n=1,2,\dots$$

the first of which, $\mathfrak{m}_1$, is a block in $\mathcal{S}$ satisfying φ at its root. Suppose now that we have already constructed a weak quasimodel $\mathfrak{m}_n$. For every quasistate $q\in Q_n-Q_{n-1}$

($Q_0 = \emptyset$) select a block $\mathfrak{b}_q \in \mathcal{S}$ with root q. Without loss of generality we may assume all the selected blocks and the weak quasimodel $\mathfrak{m}_n$ to be disjoint. The weak quasimodel $\mathfrak{m}_{n+1}$ is then the result of hooking the selected blocks $\mathfrak{b}_q$ to $\mathfrak{m}_n$ by identifying their roots q with $q \in Q_n - Q_{n-1}$.

Define the limit $\mathfrak{m} = \langle Q, T_{\alpha_1}, \ldots, T_{\alpha_k} \rangle$, of the constructed sequence by taking

$$Q = \bigcup \{Q_n : n \geq 1\}, \quad T_{\alpha_i} = \bigcup \{T^n_{\alpha_i} : n \geq 1\}.$$

It follows from the construction and the definition of a block that $\mathfrak{m}$ is a quasimodel for φ. ❑

As a consequence of this criterion we obtain the following:

Theorem 16. *The satisfiability problem for $\mathcal{PDLC}$-formulas is decidable.*

Remark 17. $\mathcal{ALC}$, the underlying concept description logic we have considered in this paper is only one representative of the extensive family of description logics (see e.g. [6, 4, 5]) that can be combined with **PDL**. And for many of them the developed technique is able to provide satisfiability checking algorithms. For instance, we can base **PDL** on the rather expressive logic $\mathcal{CIQ}$ of [4, 5] which has means for constructing inverses, unions, compositions and transitive reflexive closures of roles as well as for restricted quantification of concepts. $\mathcal{CIQ}$ does not enjoy the finite model property but is decidable, and this is enough to establish decidability of its hybrid with **PDL**.

References

[1] F. Baader and B. Hollunder. A terminological knowledge representation system with complete inference algorithms. In *Proceedings of the workshop on Processing Declarative Knowledge, PDK-91*, pages 67–86. Lecture Notes in Artificial Intelligence, No. 567. Springer Verlag, 1991.

[2] F. Baader and A. Laux. Terminological logics with modal operators. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 808–814, Montreal, Canada, 1995. Morgan Kaufman.

[3] R.J. Brachman and J.G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9:171–216, 1985.

[4] G. de Giacomo. *Decidability of Class-Based Knowledge Representation Formalisms.* PhD thesis, Univ. di Roma, 1995.

[5] G. de Giacomo and M. Lenzerini. TBox and ABox reasoning in expressive description logics. In *Proceedings of the fifth Conference on Principles of Knowledge Representation and Reasoning*, Montreal, Canada, 1996. Morgan Kaufman.

[6] F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in description logics. In G. Brewka, editor, *Principles of Knowledge Representation*, pages 191–236. CSLI Publications, 1996.

[7] D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2, pages 497–604. Reidel, Dordrecht, 1984.

[8] T.S. Kaczmarek, R. Bates, and G. Robins. Recent developments in NIKL. In *Proceedings of the 5th Nat. Conf. on Artificial Intelligence (AAAI-86)*, pages 978–985, 1986.

[9] A. Laux. Beliefs in multi-agent worlds: a terminological approach. In *Proceedings of the 11th European Conference on Artificial Intelligence*, pages 299–303, Amsterdam, 1994.

[10] J.J. Meyer. A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic*, 29:109–136, 1988.

[11] V.R. Pratt. Semantical considerations on Floyd–Hoare logic. In *Proc. 17th IEEE Symp. Found. Comput. Sci.*, pages 109–121, 1976.

[12] H. Prendinger and G. Schurz. Reasoning about action and change. *Journal of Logic, Language and Information*, 5:209–245, 1996.

[13] K. Schild. A correspondence theory for terminological logics: preliminary report. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)*, pages 466–471, Sydney, 1991.

[14] K. Schild. Combining terminological logics with tense logic. In *Proceedings of the 6th Portuguese Conference on Artificial Intelligence*, pages 105–120, Porto, 1993.

[15] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.

[16] A. Schmiedel. A temporal terminological logic. In *Proceedings of the 9th National Conference of the American Association for Artificial Intelligence*, pages 640–645, Boston, 1990.

[17] K. Segerberg. Applying modal logic. *Studia Logica*, 39:275–295, 1980.

[18] E. Spaan. *Complexity of Modal Logics.* PhD thesis, Department of Mathematics and Computer Science, University of Amsterdam, 1993.

[19] F. Wolter and M. Zakharyaschev. Satisfiability problem in description logics with modal operators. In *Proceedings of the sixth Conference on Principles of Knowledge Representation and Reasoning*, Montreal, Canada, 1998. Morgan Kaufman.

[20] F. Wolter and M. Zakharyaschev. Temporalizing description logics. Manuscript, `http://www.informatik.uni-leipzig.de/~wolter`, 1998.