MIKHAIL RYBAKOV, DMITRY SHKATOV

Undecidability of QLTL and QCTL with two variables and one monadic predicate letter

Mikhail Rybakov

Tver State University,
Zhelyabova Street, 33, Tver, 170100, Russia.
E-mail: m_rybakov@mail.ru

Dmitry Shkatov

University of the Witwatersrand, Johannesburg,
Private Bag 3, WITS2050, South Africa.
Tver State University,
Zhelyabova Street, 33, Tver, 170100, Russia.
E-mail: shkatov@gmail.com

Abstract: We study the algorithmic properties of the quantified (first-order) temporal logics **QLTL** and **QCTL** in languages with restrictions on the number of individual variables as well as the number and arity of predicate letters. We prove that satisfiability problems for **QLTL** and **QCTL** in languages with two individual variables and one monadic predicate letter are Σ_1^1 -hard. Thus, **QLTL** and **QCTL** are Π_1^1 -hard, and so not recursively enumerable, in such languages. The results, which hold both for the expanding domain and for the constant domain semantics, are obtained by reduction from a Σ_1^1 -hard $\mathbb{N} \times \mathbb{N}$ recurrent tiling problem. The results presented in this paper strengthen those by I. Hodkinson, F. Wolter, and M. Zakharyashev, who have shown that the satisfiability problem for **QLTL** (and, by a straightforward modification of the proof, for **QCTL**) is Σ_1^1 -hard in languages with two individual variables and an unlimited supply of monadic predicate letters.

Keywords: quantified temporal logic, first-order temporal logic, linear-time temporal logic, branching-time temporal logic, restricted languages, undecidability, recursive enumerability, arithmetic complexity, algorithmic classification problem, satisfiability problem, validity problem

For citation: Rybakov M., Shkatov D. “Undecidability of QLTL and QCTL with two variables and one monadic predicate letter”, *Logicheskie Issledovaniya / Logical Investigations*, 2021, Vol. 27, No. 2, pp. 93–120. DOI: 10.21146/2074-1472-2021-27-2-93-120

1. Introduction

Temporal logics [Emerson, 1990; Goldblatt, 1992; Gabbay et al., 1994; Demri et al., 2016] (we use the term “temporal logics” for formalisms studied

primarily in the context of computer science; by contrast, the term “tense logics” is commonly used for formalisms studied primarily by philosophers and linguists), having appeared in the 1980s [Pnueli, 1986] as a tool for the verification of concurrent systems, have since been extensively used as specification languages [Manna, Pnueli, 1992; Manna, Pnueli, 1995; Clarke et al., 2000], knowledge representation formalisms [Baader et al., 2015; Borgwardt et al., 2015; Bourgaux et al., 2019; Artale et al., 2004], and query languages [Chomicki, Niwinski, 1993; Chomicki, 1994; Abiteboul, 1996], often in combination with other formalisms, such as epistemic [Halpern, Vardi, 1998; Fagin et al., 1995; van der Hoek, Wooldridge, 2003] and description [Wolter, Zakharyashev, 2000; Artale, Franconi, 2000; Baader et al., 2012] logics.

Most applications of temporal logics in computer science have been those of propositional temporal logics, mostly because the quantified temporal logics of relevance to computer science are computationally badly behaved — most of them are not even recursively enumerable¹ and, hence, deductively incomplete: no effective proof system can capture all the valid inferences in quantified temporal languages [Andréka et al., 1979; Szalas, 1986; Szalas, Holenderski, 1988; Abadi, 1989; Merz, 1992; Garson, 2001]. By comparison, propositional temporal logics, which are typically PSPACE-, EXPTIME-, or 2EXPTIME-complete [Fischer, Ladner, 1979; Sistla, Clarke, 1985; Vardi, Stockmeyer, 1985; Emerson, Halpern, 1985; Goranko, van Drimmelen, 2006; Walther et al., 2006; Schewe, 2008] and usually have sound and complete proof systems [Emerson, Halpern, 1985; Goldblatt, 1992; Goranko, van Drimmelen, 2006; Goranko, Shkatov, 2009d]², look quite attractive.

In some applications, however, — temporal databases being an example — propositional logics do not suffice; hence, a need for quantified temporal logics. The bad computational behaviour of quantified temporal logics does not have to spell end to their applicability: the classical first-order logic, despite being undecidable [Church, 1936; Turing, 1936], has had a very long and highly successful history of applications in computer science and artificial intelligence. What is important from the point of view of applications is not whether a logic as a whole is decidable, but whether it has sufficiently expressive decidable fragments: no application requires the full expressive power of a logic. Therefore, the study of the algorithmic properties of *fragments* of quantified temporal

¹For a brief discussion of how this relates to their propositional bases, see the authors’ earlier article [Rybakov, Shkatov, 2020a].

²These can be used to obtain, by being combined with proof systems for other formalisms such as epistemic logics [Goranko, Shkatov, 2008; Goranko, Shkatov, 2009a; Ajspur et al., 2013], proof systems for combined systems such as temporal-epistemic logics [Goranko, Shkatov, 2009b; Goranko, Shkatov, 2009c].

logics is crucial for grasping the range of their applicability within computer science and artificial intelligence.

The study of the algorithmic properties of fragments of a logic, which typically involves identification of its maximal decidable and minimal undecidable fragments — the decidable fragments that become undecidable when slightly extended and the undecidable fragments that become decidable when slightly restricted, — is known as an *algorithmic classification problem* for a logic. Such a study pursues at least two objectives: to understand the root causes of the bad computational behaviour of a logic and to obtain expressive, yet computationally feasible, fragments. The first objective is pursued mainly through identification of undecidable, the second through identification of decidable, fragments.

The main success story of algorithmic classification research has been the classical first-order logic **QCI** [Börger et al., 1997]. The algorithmic classification problem for non-classical — modal, temporal, and superintuitionistic — logics has been much less studied: despite extensive literature [Kripke, 1962; Artemov, Dzhabaridze, 1990; Gabbay, Shehtman, 1993; Wolter, Zakharyashev, 2001; Kontchakov et al., 2005; Rybakov, Shkatov, 2019c; Shehtman, Shkatov, 2019; Rybakov, Shkatov, 2020b; Rybakov, Shkatov, 2020c; Shehtman, Shkatov, 2020; Rybakov, Shkatov, 2021b; Rybakov, Shkatov, 2021c; Rybakov, Shkatov, 2021d; Rybakov, Shkatov, 2021e], much less is known about the algorithmic properties of fragments of non-classical logics than about the algorithmic properties of fragments of **QCI**.

The present paper aims to contribute to the algorithmic classification problem for quantified temporal logics. We originally focus on the quantified linear-time temporal logic **QLTL**, both the simplest and the most widely used quantified temporal logic, and then show how our treatment of **QLTL** can be extended to the quantified branching-time temporal logic **QCTL**, and hence to the quantified alternating-time temporal logic **QATL**. Clearly, our results also apply to **QCTL*** and **QATL***, since **QCTL** can be viewed as a fragment of **QCTL*** and **QATL** as a fragment of **QATL***.

The largest known decidable fragments of **QLTL** are *monodic* fragments [Hodkinson et al., 2000, Theorem 15] (see also [Gabbay et al., 2003, Ch. 11]): a monodic fragment is obtained by adding to a decidable fragment of **QCI** valid temporal formulas in which scopes of temporal operators do not contain more than one free variable (many temporal formulas of interest, including the Barcan formula corresponding to the semantic condition of constant domains, have this form). Following up on the discovery of monodic fragments, the complexity of a number of decidable fragments of **QLTL** has been iden-

tified [Hodkinson et al., 2003], and effective axiomatizations of monodic fragments have been obtained [Wolter, Zakharyashev, 2002].

The smallest known undecidable fragment of **QLTL** [Hodkinson et al., 2000, Theorem 2] contains formulas with two individual variables and an unlimited supply of monadic predicate letters.

Thus, the gap between the known decidable and the known undecidable fragments of **QLTL** is quite narrow: the fragment with two individual variables, an unlimited supply of monadic predicate letters, and unrestricted use of temporal operators is undecidable, but the fragment with two variables, only monadic predicate letters, and the monodic use of temporal operators — i.e., using the temporal operators so that their scopes contain formulas with at most one free variable — is decidable (the corresponding fragment of **QCI** is decidable [Mortimer, 1975; Grädel et al., 1997]; hence, it forms a suitable classical basis for a monodic fragment).

Nevertheless, in light of recent results [Rybakov, Shkatov, 2019c; Rybakov, Shkatov, 2020b; Rybakov, Shkatov, 2020c; Rybakov, Shkatov, 2021b; Rybakov, Shkatov, 2021c] on undecidability of quantified modal and superintuitionistic logics, close relations of quantified temporal logics, — which show that fragments with a few variables (two [Rybakov, Shkatov, 2019c; Rybakov, Shkatov, 2021c] or three [Rybakov, Shkatov, 2020c; Rybakov, Shkatov, 2021b], depending on the logic) and a *single* monadic predicate letter often turn out to be undecidable when no restrictions are placed on the use of non-classical operators — a very natural question arises of whether the gap between the known decidable and the known undecidable fragments can be tightened still further: can we prove undecidability of **QLTL** in languages with two variables and a single monadic predicate letter?

In the present paper we do just that: we narrow the gap between the known decidable and the known undecidable fragments of **QLTL** by proving that the fragment containing formulas with two variables and a *single* monadic predicate letter is already undecidable — more precisely, this fragment is Π_1^1 -hard, i.e. has the same computational complexity as the full logic (we are not aware of upper-bound results for **QLTL**; whatever the upper bound is, it is surely inherited by all the fragments).

The main technical novelty of the present paper is modelling of an arbitrary number of monadic predicate letters with a single one for logics of linear structures. The techniques used in the proofs of analogous results for most modal and superintuitionistic logics [Rybakov, Shkatov, 2019c; Rybakov, Shkatov, 2020c; Rybakov, Shkatov, 2021b] are inapplicable to **QLTL**: being ultimately based on the propositional level techniques [Halpern, 1995; Chagrov, Rybakov, 2003; Rybakov, Shkatov, 2018a; Rybakov, Shkatov, 2018b; Rybakov,

Shkatov, 2019a; Rybakov, Shkatov, 2021a] developed for modal logics of frames without restrictions on the branching factor, they cannot be used for logics of linear frames, such as **QLTL**, since they rely on transformations of models that arbitrarily increase their branching factor. We adopt here to **QLTL** an approach recently [Rybakov, Shkatov, 2020b; Rybakov, Shkatov, 2021c] used for monomodal logics of linear frames: in that setting, however, it is not clear how undecidability can be obtained for languages with a single monadic predicate letter. In the present paper, we adapt those techniques to **QLTL** to obtain undecidability for languages with two variables and a single monadic predicate letter.³

The proof proceeds by reduction from the Σ_1^1 -hard [Harel, 1986] $\mathbb{N} \times \mathbb{N}$ recurrent tiling problem. The initial reduction to satisfiability for **QLTL** in languages with two individual variables, one binary, and an unlimited supply of monadic letters, carried out in Lemma 1, resembles the reduction from a known proof by I. Hodkinson, F. Wolter, and M. Zakharyashev [Hodkinson et al., 2000, Theorem 2]; it is not, however, enough for our purposes to reuse their formulas — we, therefore, define our own reduction from the recurrent tiling problem preparing the ground for subsequent reductions to less expressive subfragments, carried out in Lemmas 2 and 3.

The paper is structured as follows. In Section 2, we introduce the syntax and semantics of **QLTL**. In Section 3, we present our undecidability result for **QLTL**. In Section 4, we show how to obtain an analogous result for **QCTL**. We conclude in Section 5 by discussing directions for future study.

We assume the reader’s familiarity with the basic notions of computability theory; the reader wishing a reminder may consult [Rogers, 1967] and [Ender-ton, 2011].

Throughout the paper, we denote by $\mathbb{N}$ the set of natural numbers, which includes zero, and by $\mathbb{N}^+$ the set of positive integers, which excludes zero.

2. Preliminaries

In this section, we present the syntax and semantics of first-order quantified linear time temporal logic **QLTL**.

2.1. Syntax of **QLTL**

The full, unrestricted, language of **QLTL** contains:

- countably many individual variables, denoted by $x, y, x_1, y_1, \dots$;
- for every $n \in \mathbb{N}$, countably many n -ary predicate letters, denoted by $P^n, Q^n, R^n, \dots$; when the arity of a predicate letter is clear from the

³The result itself was announced, without proof, in the Discussion section of an earlier paper [Rybakov, Shkatov, 2021c].

context, the superscript is usually omitted; nullary predicate letters are propositional variables;

- the Boolean constant $\perp$ (falsity);
- the binary Boolean connective $\rightarrow$ (implication);
- the binary temporal connective $\mathcal{U}$ (until);
- the quantifier symbol $\forall$ (for every);
- technical symbols: parentheses $(,)$, and the comma.

Formulas are defined by the BNF expression

$$\varphi := P(x_1, \dots, x_n) \mid \perp \mid (\varphi \rightarrow \varphi) \mid (\varphi \mathcal{U} \varphi) \mid \forall x \varphi,$$

where P ranges over n -ary predicate letters. *Atomic* formulas are those of the form $P(x_1, \dots, x_n)$. *Closed* formulas are those without free occurrences of variables. We use the standard abbreviations

$$\begin{aligned} \neg \varphi &= (\varphi \rightarrow \perp); \\ \top &= \neg \perp; \\ \varphi \wedge \psi &= \neg \varphi \rightarrow \neg \psi; \\ \varphi \vee \psi &= \neg \varphi \rightarrow \psi; \\ \varphi \leftrightarrow \psi &= (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi); \\ \bigcirc \varphi &= \perp \mathcal{U} \varphi; \\ \Diamond \varphi &= \varphi \vee (\top \mathcal{U} \varphi); \\ \Box \varphi &= \neg \Diamond \neg \varphi; \\ \exists x \varphi &= \neg \forall x \neg \varphi, \end{aligned}$$

as well as

$$\begin{aligned} \bigcirc^0 \varphi &= \varphi; \\ \bigcirc^{n+1} \varphi &= \bigcirc \bigcirc^n \varphi, \text{ for every } n \in \mathbb{N}. \end{aligned}$$

When parentheses are omitted, $\neg$, $\bigcirc$, $\Diamond$, $\Box$, $\forall$, and $\exists$ are assumed to bind tighter than $\wedge$, $\vee$, and $\mathcal{U}$, which are assumed to bind tighter than $\rightarrow$ and $\leftrightarrow$. To enhance readability of formulas, we occasionally use square brackets in place of parentheses.

2.2. Semantics of QLTL

The semantics of **QLTL**-formulas is defined in terms of *Kripke frames*, also known — particularly, in computer science — as *transition systems*.

A *Kripke frame* is a structure $\mathfrak{F} = \langle S, \mapsto \rangle$, where S is a non-empty set of *states* and $\mapsto$ is a binary *transition relation* on S satisfying the seriality condition: for every $s \in S$, there exists $t \in S$ such that $s \mapsto t$. If $s \mapsto t$, we say that t is *accessible from* s . The relation $\mapsto$ is not required to be functional: more than one state can be accessible from any state.

A *path* in a frame $\langle S, \mapsto \rangle$ is an infinite sequence π of elements of S such that $\pi[i] \mapsto \pi[i+1]$, for every $i \in \mathbb{N}$; thus, $\pi[i]$ is the i th state of π . For every $i \in \mathbb{N}$, we denote by $\pi[i, \infty]$ the infinite suffix of π beginning with $\pi[i]$, i.e., the sequence $\pi[i], \pi[i+1], \dots$ of states.

A *predicate Kripke frame with a system of expanding domains* is a tuple $\mathfrak{F}_D = \langle S, \mapsto, D \rangle$, where $\langle S, \mapsto \rangle$ is a Kripke frame and D is a function from S into the set of non-empty subsets of some set, *the domain of* $\mathfrak{F}_D$; the function D is required to satisfy the monotonicity condition: $s \mapsto s'$ implies $D(s) \subseteq D(s')$. The set $D(s)$ is *the domain of state* s . We often write D_s for $D(s)$. We shall also consider predicate frames satisfying the stronger constancy condition: $s \mapsto s'$ implies $D(s) = D(s')$; such predicate frames are called *predicate frames with constant domains*. By *predicate frame simpliciter* we mean predicate frames with expanding domains. If needed, consult [Gabbay et al., 1994; Hughes, Cresswell, 1996; Fitting, Mendelsohn, 1998; Braüner, Ghilardi, 2007; Goldblatt, 2011] and [Gabbay et al., 2009, §3.1] for fuller description of Kripke semantics for predicate temporal — and closely related modal — logics.

A *Kripke model* is a tuple $\mathfrak{M} = \langle S, \mapsto, D, I \rangle$, where $\langle S, \mapsto, D \rangle$ is a predicate Kripke frame and I , called *the interpretation of predicate letters* with respect to states of S , is a function assigning to a state $s \in S$ and an n -ary predicate letter P an n -ary relation $I(s, P)$ on D_s ; thus, $I(s, P) \subseteq D_s^n$. In particular, if P is nullary, $I(s, P) \subseteq D_s^0 = \{\langle \rangle\}$; thus, we can identify truth with $\{\langle \rangle\}$ and falsity with $\emptyset$. We often write $P^{I,s}$ for $I(s, P)$. We say that $\langle S, \mapsto, D, I \rangle$ is a model *over* the frame $\langle S, \mapsto \rangle$ and *over* the predicate frame $\langle S, \mapsto, D \rangle$. We also say that $\langle S, \mapsto \rangle$ is the *underlying frame*, and $\langle S, \mapsto, D \rangle$ the *underlying predicate frame*, of a model $\langle S, \mapsto, D, I \rangle$. If π is a path in a frame underlying a model $\mathfrak{M}$, we also say that π is a path in $\mathfrak{M}$.

An *assignment* in a model is a function g associating with every individual variable x an element $g(x)$ of the domain of the underlying predicate frame. We write $g' \stackrel{x}{=} g$ if assignment g' differs from assignment g in at most the value of variable x .

Formulas are evaluated in Kripke models relative to paths (rather than states, as in modal, and some other temporal, logics). The satisfaction relation between models $\mathfrak{M}$, paths π , assignments g , and formulas φ is defined by recursion:

- $\mathfrak{M}, \pi \models^g P(x_1, \dots, x_n) \iff \langle g(x_1), \dots, g(x_n) \rangle \in P^{I, \pi[0]}$;
- $\mathfrak{M}, \pi \not\models^g \perp$;
- $\mathfrak{M}, \pi \models^g \varphi_1 \rightarrow \varphi_2 \iff \mathfrak{M}, \pi \not\models^g \varphi_1 \text{ or } \mathfrak{M}, \pi \models^g \varphi_2$;

- $\mathfrak{M}, \pi \models \varphi_1 \mathcal{U} \varphi_2 \iff \mathfrak{M}, \pi[i, \infty] \models \varphi_2$, for some $i > 0$, and $\mathfrak{M}, \pi[j, \infty] \models \varphi_1$, for every j with $0 < j < i$;
- $\mathfrak{M}, \pi \models^g \forall x \varphi_1 \iff \mathfrak{M}, \pi \models^{g'} \varphi_1$, for every g' such that $g' \stackrel{x}{=} g$ and $g'(x) \in D(\pi[0])$.

This definition implies that

- $\mathfrak{M}, \pi \models^g \bigcirc \varphi \iff \mathfrak{M}, \pi[1, \infty] \models^g \varphi$;
- $\mathfrak{M}, \pi \models^g \Diamond \varphi \iff \mathfrak{M}, \pi[i, \infty] \models^g \varphi$, for some $i \in \mathbb{N}$;
- $\mathfrak{M}, \pi \models^g \Box \varphi \iff \mathfrak{M}, \pi[i, \infty] \models^g \varphi$, for every $i \in \mathbb{N}$.

We note that, if $\mathfrak{M} = \langle S, \mapsto, D, I \rangle$ is a Kripke model, $s \in S$, and $I_s(P) = I(s, P)$, then the tuple $\mathfrak{M}_s = \langle D_s, I_s \rangle$ is a classical predicate model, or structure.

We also note that we take the strict version of “until” as our primitive temporal connective: $\pi \models \varphi_1 \mathcal{U} \varphi_2$ implies that φ_2 is satisfied at a future state of π and φ_1 is satisfied at every future state of π preceding the state satisfying φ_2 . The non-strict “until”, allowing φ_2 to be satisfied at the current state of π , can be expressed using the strict “until”: $\varphi_1 \mathcal{U}^\circ \varphi_2 = \varphi_2 \vee (\varphi_1 \mathcal{U} \varphi_2)$. Conversely, the strict until can be expressed using $\mathcal{U}^\circ$ and $\bigcirc$: $\varphi_1 \mathcal{U} \varphi_2 = \bigcirc(\varphi_1 \mathcal{U}^\circ \varphi_2)$. Thus, it is immaterial whether $\{\bigcirc, \mathcal{U}^\circ\}$ or $\{\mathcal{U}\}$ is taken as the set of primitive temporal operators.

We shall often use the following notation. Let $\mathfrak{M} = \langle S, \mapsto, D, I \rangle$ be a Kripke model, π a path in $\mathfrak{M}$, and $a_1, \dots, a_n$ elements of $D(\pi[0])$; let also $\varphi(x_1, \dots, x_n)$ be a formula whose free variables are among $x_1, \dots, x_n$ and g be an assignment with $g(x_1) = a_1, \dots, g(x_n) = a_n$. Then, we shall often write $\mathfrak{M}, \pi \models \varphi(a_1, \dots, a_n)$ instead of $\mathfrak{M}, \pi \models^g \varphi(x_1, \dots, x_n)$. This notation is unambiguous since the languages we consider do not contain constants.

Since the satisfaction of atomic formulas at a path depends only on its initial state, we often write $\mathfrak{M}, s \models P(a_1, \dots, a_n)$ instead of $\mathfrak{M}, \pi \models P(a_1, \dots, a_n)$ whenever $\pi[0] = s$. If $\mathfrak{M}, s \models P(a_1, \dots, a_n)$, we also say that $P(a_1, \dots, a_n)$ is true at s in $\mathfrak{M}$.

A formula φ is *satisfied at a path* π in a model $\mathfrak{M}$ (symbolically, $\mathfrak{M}, \pi \models \varphi$) if $\mathfrak{M}, \pi \models^g \varphi$, for every g assigning to the free variables of φ elements of $D(\pi[0])$. A formula φ is *satisfied at a state* s of a model $\mathfrak{M}$ (symbolically, $\mathfrak{M}, s \models \varphi$) if $\mathfrak{M}, \pi \models \varphi$, for every path π with $\pi[0] = s$. A formula φ is *true in a model* $\mathfrak{M}$ (symbolically, $\mathfrak{M} \models \varphi$) if $\mathfrak{M}, s \models \varphi$, for every state s of $\mathfrak{M}$. A formula φ is *valid on a predicate frame* $\mathfrak{F}_D$ (symbolically, $\mathfrak{F}_D \models \varphi$) if φ is true in every model over $\mathfrak{F}_D$. A formula φ is *valid on a frame* $\mathfrak{F}$ (symbolically, $\mathfrak{F} \models \varphi$) if φ is valid on every predicate frame $\langle \mathfrak{F}, D \rangle$.

To simplify the notation, we write $\pi \models \varphi$ and $s \models \varphi$, rather than, respectively, $\mathfrak{M}, \pi \models \varphi$ and $\mathfrak{M}, s \models \varphi$, when $\mathfrak{M}$ is clear from the context.

A formula φ is *satisfiable in a model* $\mathfrak{M} = \langle S, \mapsto, D, I \rangle$ if there exists a path π in $\mathfrak{M}$ and an assignment g assigning to the free variables of φ elements of $D(\pi[0])$ such that $\mathfrak{M}, \pi \models^g \varphi$.

We note that satisfiability of a closed formula in a model does not depend on the assignment: a closed formula φ is satisfiable in a model $\mathfrak{M} = \langle S, \mapsto, D, I \rangle$ if, and only if, there exists a path π in $\mathfrak{M}$ such that $\mathfrak{M}, \pi \models \varphi$.

A formula φ is **QLTL**-*valid* if it is valid on every frame; it is **QLTL**-*satisfiable* if it is satisfiable in some model.

In this paper, we are concerned with the computational complexity of the following computational problems:

- *Validity problem for QLTL*: given a formula φ , determine whether φ is **QLTL**-valid.
- *Satisfiability problem for QLTL*: given a formula φ , determine whether φ is **QLTL**-satisfiable.

We note that φ is valid if, and only if, $\neg\varphi$ is not satisfiable; therefore, validity and satisfiability for **QLTL** are dual problems.

3. Main results

In this section, we prove that satisfiability for **QLTL** is Σ_1^1 -hard — hence, validity for **QLTL** is Π_1^1 -hard — in languages with two individual variables and a single monadic predicate letter. This implies that the set of validities of **QLTL** in such languages is not recursively enumerable, and hence cannot be captured by an effective sound and complete proof system.

3.1. Reduction from the recurrent tiling problem

We proceed by reduction from a Σ_1^1 -complete [Harel, 1986] recurrent tiling problem for $\mathbb{N} \times \mathbb{N}$.⁴ We are given a set of tiles, a tile t being a 1×1 square, with a fixed orientation, whose edges are “colored” with $left(t)$, $right(t)$, $up(t)$, and $down(t)$. A tile type is fully determined by the edge colors. Every tile belongs to one of the finitely many types $T = \{t_0, \dots, t_\alpha\}$, with tiles of each type being in unlimited supply. A *tiling* in an arrangement of tiles on a rectangular $\mathbb{N} \times \mathbb{N}$ grid so that the edge colors of the adjacent tiles match, both horizontally and vertically. We are to determine whether there exists a tiling in which a tile of

⁴D. Harel proves Σ_1^1 -completeness of this problem by one-to-one reduction from a Σ_1^1 -complete set of nondeterministic Turing machines: each machine is mapped to an instance of the recurrent tiling problem, i.e., to a set of tile types.

type t_0 occurs infinitely often in the leftmost column, i.e., whether there exists a function $f : \mathbb{N} \times \mathbb{N} \rightarrow T$ such that, for every $n, m \in \mathbb{N}$,

$$(T_1) \text{ } right(f(n, m)) = left(f(n + 1, m));$$

$$(T_2) \text{ } up(f(n, m)) = down(f(n, m + 1));$$

$$(T_3) \text{ the set } \{m \in \mathbb{N} : f(0, m) = t_0\} \text{ is infinite.}$$

To make the underlying idea clearer, we initially encode the recurrent tiling problem with formulas of two individual variables, without regard for the number of predicate letters used. This encoding is similar to the one used by I. Hodkinson, F. Wolter, and M. Zakharyashev [Hodkinson et al., 2000, Theorem 2] (similar techniques have been used elsewhere [Spaan, 1993; Marx, 1999; Wolter, Zakharyashev, 2001; Kontchakov et al., 2005] and [Gabbay et al., 2003, Theorem 11.1]), but our encoding is different from theirs since we need to prepare the ground for the subsequent reductions, presented in Lemmas 2 and 3, eliminating all the predicate letter bar a single monadic one in the formula obtained in the initial encoding. Thus, the formulas we use in the initial encoding resemble, but are not the same as, those used by I. Hodkinson, F. Wolter, and M. Zakharyashev [Hodkinson et al., 2000, Theorem 2].

To obtain a grid for the tiling, we shall think of the states of the path satisfying a formula as rows, and of elements of the domain of the state satisfying a formula as columns, of the $\mathbb{N} \times \mathbb{N}$ grid. Indeed, a path can be thought of as the set of natural numbers with the immediate successor relation; we shall also write formulas that capture, within the domain of a state, the structure of the naturals with the immediate successor relation. Then, to describe a tiling on thus obtained grid, we use a family $\{P_t : t \in T\}$ of monadic predicate letters corresponding to the tile types: intuitively, $\pi[m] \models P_t(a_n)$ shall mean that the square $\langle n, m \rangle$ is tiled with a tile of type t from T .

Not all the states on the satisfying path shall be part of the tiling: to prepare the ground for subsequent reductions, we create gaps, non-empty equal-sized sequences of states that are not part of the tiling, in the timeline; the states meant to partake in the tiling shall be marked off by a propositional variable p . The size of the gaps, $\alpha + 6$, is proportional to the number of monadic predicate letters we need to eliminate in subsequent reductions.

Thus, let $\triangleleft$ be a binary, P_t , for every $t \in T$, monadic, and p nullary predicate letter.

Let (for brevity, in formulas we write l , r , u , and d instead of *left*, *right*, *up*, and *down*)

$$\begin{aligned}
 A_0 &= p \wedge \Box(p \leftrightarrow \bigcirc^{\alpha+6} p); \\
 A_1 &= \exists x \Box \Diamond(p \wedge P_{t_0}(x)); \\
 A_2 &= \forall x \exists y x \triangleleft y; \\
 A_3 &= \forall x \forall y [x \triangleleft y \rightarrow \Box(p \rightarrow x \triangleleft y)]; \\
 A_4 &= \forall x \Box [p \rightarrow (\bigvee_{t \in T} P_t(x) \wedge \bigwedge_{t' \neq t} (P_t(x) \rightarrow \neg P_{t'}(x)))]; \\
 A_5 &= \forall x \forall y \Box \bigwedge_{t \in T} (x \triangleleft y \wedge P_t(x) \rightarrow \bigvee_{r(t)=l(t')} P_{t'}(y)); \\
 A_6 &= \forall x \Box \bigwedge_{t \in T} (P_t(x) \rightarrow \bigcirc^{\alpha+6} \bigvee_{u(t)=d(t')} P_{t'}(x)),
 \end{aligned}$$

and let A be the conjunction of A_0 through A_6 . Notice that the formula A contains only two individual variables.

Fig. 1. Model $\mathfrak{M}_0$

Lemma 1. *There exists a recurrent tiling of $\mathbb{N} \times \mathbb{N}$ satisfying (T_1) through (T_3) if and only if A is satisfiable.*

Proof. (“if”) Suppose A is satisfiable, i.e., $\mathfrak{M}, \pi \models A$, for some model $\mathfrak{M} = \langle S, \mapsto, D, I \rangle$ and some path π in $\mathfrak{M}$.

By A_0 ,

$$\pi[m] \models p \iff m \equiv 0 \pmod{\alpha + 6}. \quad (1)$$

By A_1 , there exists $a_0 \in D(\pi[0])$ such that $\pi[0] \models P_{t_0}(a_0)$.

By A_2 , there exists an infinite sequence $a_0, a_1, a_2, \dots$ of elements of $D(\pi[0])$ such that

$$a_0 \triangleleft^{I, \pi[0]} a_1 \triangleleft^{I, \pi[0]} a_2 \triangleleft^{I, \pi[0]} \dots$$

By A_3 and (1), for every m such that $m \equiv 0 \pmod{\alpha + 6}$,

$$a_0 \triangleleft^{I, \pi[m]} a_1 \triangleleft^{I, \pi[m]} a_2 \triangleleft^{I, \pi[m]} \dots$$

By A_4 , for every $n \in \mathbb{N}$ and every m such that $m \equiv 0 \pmod{\alpha + 6}$, there exists a unique $t \in T$ such that $\pi[m] \models P_t(a_n)$. Hence, we can define a function $f : \mathbb{N} \times \mathbb{N} \rightarrow T$ by

$$f(n, m) = t \text{ whenever } \pi[m(\alpha + 6)] \models P_t(a_n).$$

Since $\pi \models A_1 \wedge A_5 \wedge A_6$, conditions (T_1) through (T_3) are satisfied for f . Therefore, f is a required function.

(“only if”) Suppose f is a function satisfying conditions (T_1) through (T_3) . We obtain a model satisfying A . Let

- $S = \mathbb{N}$,
- $s \mapsto s' \iff s' = s + 1$,
- $D(n) = \mathbb{N}$, for every $n \in \mathbb{N}$,

and let $\mathfrak{M}_0 = \langle \mathbb{N}, <, D, I \rangle$ be a model, depicted in Figure 1, such that, for every $n \in \mathbb{N}$,

- $n \models p \iff n \equiv 0 \pmod{\alpha + 6}$;
- $n \models k \triangleleft l \iff n \equiv 0 \pmod{\alpha + 6} \text{ and } l = k + 1$;
- $n \models P_t(k) \iff \exists m \in \mathbb{N} (n = m(\alpha + 6) \text{ and } f(m, k) = t)$.

Lastly, let π_0 be the path defined by $\pi_0[n] = n$, for every $n \in \mathbb{N}$.

It is straightforward to check that $\mathfrak{M}_0, \pi_0 \models A$, so we leave this to the reader. ■

Thus, in the proof of the “if” part of Lemma 1, we obtained a grid for the required tiling by treating the states of path π as rows and elements $a_0, a_1, a_2, \dots$ of the domain of the state $\pi[0]$ as columns.

3.2. Elimination of the binary predicate letter

We next eliminate, in a satisfiability-preserving way, all predicate letters of the formula A bar a single monadic one, without increasing the number of individual variables in the resultant formula. We, thus, obtain a reduction from the recurrent tiling problem to satisfiability for **QLTL** in languages with two variables and a single monadic predicate letter.

The elimination of predicate letters is carried out in two steps: first, we simulate the binary letter $\triangleleft$ with two monadic predicate letters; then, we simulate all the predicate letters of thus obtained formula with a single monadic one.

From now on, we assume, for ease of notation, that the formula A contains monadic predicate letters $P_0, \dots, P_\alpha$ — rather than a letters P_t , for each $t \in \{t_0, \dots, t_\alpha\}$ — to refer to the tile types.

First, using a suitable adaptation of the ideas of Kripke's [Kripke, 1962], we eliminate the binary predicate letter $\triangleleft$ of A .

Recall that Kripke's construction [Ibid.] transforms a model $\mathfrak{M}$ satisfying a formula containing a binary predicate letter, and no modal operators, at a state s in such a way that a sufficiently large number of states — more precisely, one state for every pair of elements of the domain of s — accessible from s is added to $\mathfrak{M}$. This idea cannot be applied to **QLTL** in a straightforward manner, for two reasons. First, in Kripke's version, the added states do not form a path, which is needed for **QLTL**. Second, since $\triangleleft$ occurs in the scope of the temporal operator $\Box$ in A , we need to simulate $\triangleleft$ not just at the initial state of a path satisfying A , but at every state affecting the satisfaction of A .

We resolve these difficulties by working the model $\mathfrak{M}_0$ defined in the proof of Lemma 1 and by exploiting its properties: the domains of the states of $\mathfrak{M}_0$ are all equal, and the interpretation of $\triangleleft$ is the same at every state of $\mathfrak{M}_0$ affecting the satisfaction of A .

Let $P_{\alpha+1}$ and $P_{\alpha+2}$ be monadic predicate letters distinct from $P_0, \dots, P_\alpha$ and from each other, and let $\cdot'$ be the function substituting

$$\Diamond(P_{\alpha+1}(x) \wedge P_{\alpha+2}(y)), \quad \text{for } x \triangleleft y.$$

Lemma 2. *There exists a recurrent tiling of $\mathbb{N} \times \mathbb{N}$ satisfying (T_1) through (T_3) if and only if A' is satisfiable.*

Proof. (“if”) Suppose $\mathfrak{M}, \pi \models A'$, for some model $\mathfrak{M} = \langle S, \mapsto, D, I \rangle$ and some path π . Define an interpretation I' on the frame $\langle S, \mapsto, D \rangle$ so that, for every $s \in S$,

$$I'(s, \triangleleft) = \{ \langle a, b \rangle : \mathfrak{M}, s \models \Diamond(P_{\alpha+1}(a) \wedge P_{\alpha+2}(b)) \}$$

and, for every $s \in S$ and every $Q \in \{p, P_0, \dots, P_\alpha\}$,

$$I'(s, Q) = I(s, Q).$$

and let $\mathfrak{M}' = \langle S, \mapsto, D, I' \rangle$. Then, $\mathfrak{M}', \pi \models A$. Hence, by Lemma 1, there exists a recurrent tiling of $\mathbb{N} \times \mathbb{N}$ satisfying (T_1) through (T_3) .

(“only if”) Suppose f is a function satisfying conditions (T_1) through (T_3) . Let $\mathfrak{M}_0$ and π_0 be, respectively, the model and the path defined in the proof of the “only if” part of Lemma 1. As we have seen, $\mathfrak{M}_0, \pi_0 \models A$. We use $\mathfrak{M}_0$ to obtain a model satisfying A' .

Let σ be the infinite sequence

$$0, 0, 1, 0, 1, 2, 0, 1, 2, 3, 0, 1, 2, 3, 4, \dots$$

and let σ_k , for each $k \in \mathbb{N}$, be the k th element of σ .

Let $\mathfrak{M}'_0 = \langle \mathbb{N}, \mapsto, D, I' \rangle$ be a model such that, for every $n, c \in \mathbb{N}$,

$$\begin{aligned} \mathfrak{M}'_0, n \models P_{\alpha+1}(c) &\iff \exists m \in \mathbb{N} (n = m(\alpha + 6) \text{ and } c = \sigma_m); \\ \mathfrak{M}'_0, n \models P_{\alpha+2}(c) &\iff \exists m \in \mathbb{N} (n = m(\alpha + 6) \text{ and } c = \sigma_m + 1), \end{aligned}$$

and for every $n \in \mathbb{N}$ and every $S \in \{p, P_0, \dots, P_\alpha\}$,

$$I'(n, S) = I(n, S).$$

We show that $\mathfrak{M}'_0, \pi_0[0] \models A'$.

Since $\mathfrak{M}_0, \pi_0[0] \models A$, it suffices to prove, for every $m, a, b \in \mathbb{N}$,

$$\mathfrak{M}_0, m(\alpha + 6) \models a \triangleleft b \iff \mathfrak{M}'_0, m(\alpha + 6) \models \Diamond(P_{\alpha+1}(a) \wedge P_{\alpha+2}(b)).$$

Assume $\mathfrak{M}_0, m(\alpha + 6) \models a \triangleleft b$. Then, $b = a + 1$, by definition of $\mathfrak{M}_0$. Choose $k \in \mathbb{N}$ so that $m \leq k$ and $\sigma_k = a$; by definition of σ , such a number k certainly exists. Then, $\mathfrak{M}'_0, k(\alpha + 6) \models P_{\alpha+1}(a) \wedge P_{\alpha+2}(b)$, by definition of $\mathfrak{M}'_0$. Hence, $\mathfrak{M}'_0, m(\alpha + 6) \models \Diamond(P_{\alpha+1}(a) \wedge P_{\alpha+2}(b))$.

Conversely, assume $\mathfrak{M}'_0, m(\alpha + 6) \models \Diamond(P_{\alpha+1}(a) \wedge P_{\alpha+2}(b))$, i.e., $\mathfrak{M}'_0, v \models P_{\alpha+1}(a) \wedge P_{\alpha+2}(b)$, for some $v \geq m(\alpha + 6)$. The definition of $\mathfrak{M}'_0$ implies that $v = k(\alpha + 6)$, for some $k \geq m$. Hence, by definition of $\mathfrak{M}'_0$, both $a = \sigma_k$ and $b = \sigma_k + 1$. Hence, $b = a + 1$. Therefore, $\mathfrak{M}_0, m(\alpha + 6) \models a \triangleleft b$, by definition of $\mathfrak{M}_0$. ■

 Fig. 2. Model $\mathfrak{M}_0^*$

3.3. Elimination of monadic predicate letters

We lastly simulate the occurrences of predicate letters $p, P_0, \dots, P_{\alpha+2}$ in A' with a single monadic letter P , without increasing the number of individual variables in the resultant formula. We, thus, obtain a reduction of the recurrent tiling problem to satisfiability for **QLTL** with formulas containing two individual variables and a single monadic predicate letter P .

Let P be a monadic predicate letter distinct from the letters $P_0, \dots, P_{\alpha+2}$. Define

$$\theta = \forall x \neg P(x) \wedge \bigcirc \forall x P(x) \wedge \bigcirc^2 \forall x P(x),$$

and, for every $k \in \{0, \dots, \alpha+2\}$,

$$\begin{aligned} \zeta_k(x) &= \theta \wedge \bigcirc^{\alpha+5-k} P(x); \\ \zeta_k(y) &= \theta \wedge \bigcirc^{\alpha+5-k} P(y). \end{aligned}$$

Let $\cdot^*$ be the function replacing

- p with θ ;
- $P_k(x)$ with $\zeta_k(x)$, for each $k \in \{0, \dots, \alpha + 2\}$;
- $P_k(y)$ with $\zeta_k(y)$, for each $k \in \{0, \dots, \alpha + 2\}$,

and let A^* be the result of applying the function $\cdot^*$ to A' .

To obtain a model satisfying A^* , provided a recurrent tiling exists, we use the model $\mathfrak{M}'_0$ defined in the proof of Lemma 2. Call a state s of $\mathfrak{M}'_0$ a *tiling* state if $\mathfrak{M}'_0, s \models p$ (recall that these are the states corresponding to the tiling) and a *gap* state otherwise. We use gap states to simulate the interpretation of letters $p, P_0, \dots, P_{\alpha+2}$ at a tiling state of $\mathfrak{M}'_0$. The truth of $P_k(a)$, for every $k \in \{0, \dots, \alpha + 2\}$ and every $a \in \mathbb{N}$, at a tiling state s is simulated by making $P(a)$ true at the gap state $k + 1$ steps away from the next tiling state. The interpretation of the nullary letter p at a tiling state s is simulated by making P universally true at the two gap states immediately succeeding s . As we shall show, this pattern of two consecutive states making P universally true does not occur anywhere else — hence, it marks off exactly the tiling states.

Lemma 3. *There exists a recurrent tiling of $\mathbb{N} \times \mathbb{N}$ satisfying (T_1) through (T_3) if and only if A^* is satisfiable in **QLTL**.*

Proof. (“if”) Suppose $\mathfrak{M}, \pi \models A^*$, for some model $\mathfrak{M} = \langle S, \mapsto, D, I \rangle$ and some path π . Define an interpretation I' on the frame $\langle S, \mapsto, D \rangle$ so that, for every $s \in S$ and every $k \in \{0, \dots, \alpha + 2\}$,

$$I'(s, P_k) = \{ \langle a \rangle : \mathfrak{M}, s \models \zeta_k(a) \}$$

and, for every $s \in S$,

$$I'(s, p) = \begin{cases} \{ \langle \rangle \} & \text{if } \mathfrak{M}, s \models \theta; \\ \emptyset & \text{if } \mathfrak{M}, s \not\models \theta, \end{cases}$$

and let $\mathfrak{M}' = \langle S, \mapsto, D, I' \rangle$. Then, $\mathfrak{M}', \pi \models A'$. Hence, by Lemma 2, there exists a recurrent tiling of $\mathbb{N} \times \mathbb{N}$ satisfying (T_1) through (T_3) .

(“only if”) Suppose f is a function satisfying conditions (T_1) through (T_3) . Let $\mathfrak{M}'_0$ and π_0 be, respectively, the model and the path defined in the “only if” part of the proof of Lemma 2. As we have seen, $\mathfrak{M}'_0, \pi_0[0] \models A'$. We use $\mathfrak{M}'_0$ to obtain a model satisfying A^* .

Let $\langle \mathbb{N}, \mapsto, D \rangle$ be the frame underlying $\mathfrak{M}'_0$, i.e. let $D(s) = \mathbb{N}$, for every $s \in \mathbb{N}$. Define a model $\mathfrak{M}^*_0 = \langle \mathbb{N}, \mapsto, D, I^* \rangle$ so that, for every $s \in \mathbb{N}$,

- if $s \equiv 0 \pmod{\alpha + 6}$, then $I^*(s, P) = \emptyset$;
- if $s \equiv 1 \pmod{\alpha + 6}$ or $s \equiv 2 \pmod{\alpha + 6}$, then $I^*(s, P) = \mathbb{N}$;
- if $s = j + n(\alpha + 6)$, for some $j \in \{3, \dots, \alpha + 5\}$ and some $n \in \mathbb{N}$, then for every $a \in \mathbb{N}$,

$$\mathfrak{M}_0^*, s \models P(a) \iff \mathfrak{M}'_0, n(\alpha + 6) \models P_{\alpha+5-j}(a).$$

Let π_0^* be a path with $\pi_0^*[0] = s_0$. It is straightforward to check that $\mathfrak{M}_0^*, \pi_0^* \models A^*$, so we leave this to the reader.

Thus, A^* is satisfiable. ■

We, therefore, obtain the following:

Theorem 1. *Satisfiability for QLTL is Σ_1^1 -hard in languages with two individual variables and a single monadic predicate letter.*

Proof. Immediate from Lemma 3. ■

As we have observed in Section 2, satisfiability and validity for QLTL are dual problems. Hence, we have also proven the following:

Corollary 1. *Validity for QLTL is Π_1^1 -hard in languages with two individual variables and a single monadic predicate letter.*

Thus, QLTL is not recursively enumerable even in languages with only two variables and only a single monadic predicate letter. Therefore, even for such a fragment of QLTL, no effective sound and complete proof system exists.

In the proofs of this section, we have never relied on the domains of the predicate frames we have worked with being properly increasing. We have, therefore, also established the following:

Theorem 2. *Satisfiability for QLTL over predicate frames with constant domains is Σ_1^1 -hard in languages with two individual variables and a single monadic predicate letter. Validity for QLTL over predicate frames with constant domains is Π_1^1 -hard in languages with two individual variables and a single monadic predicate letter.*

4. QCTL and QATL

Since **QCTL** is neither more nor less expressive than **QLTL**, an undecidability result for either of these logics does not automatically imply an analogous result for the other logic. The formulas used in Section 3 were, however, chosen so that a simple translation, presented in this section, effectively embeds the undecidable fragment of **QLTL** obtained in Section 3 into a fragment of **QCTL**. Since the translation does not affect either the individual variables or the predicate letters of a formula, we obtain a result analogous to Theorem 2 for **QCTL**.

Recall that the main difference between **QCTL** and **QLTL** (just as between their propositional bases **CTL** and **LTL** [Huth, Ryan, 2004]) is that in **QCTL** formulas are evaluated at states rather than at paths, as in **QLTL**, of Kripke models. The language of **QCTL** contains, in place of formulas $(\varphi_1 \mathcal{U} \varphi_2)$, formulas $\mathcal{E}(\varphi_1 \mathcal{U} \varphi_2)$ and $\mathcal{A}(\varphi_1 \mathcal{U} \varphi_2)$ (“existential until” and “universal until”, respectively; the quantifiers $\mathcal{E}$ and $\mathcal{A}$ refer to the paths beginning with the state at which the formula is evaluated). The satisfaction conditions for these formulas are (here, s is a state of a Kripke model)

- $\mathfrak{M}, s \models \mathcal{E}(\varphi_1 \mathcal{U} \varphi_2) \Leftrightarrow$ there exists a path π with $\pi[0] = s$ such that $\mathfrak{M}, \pi[i] \models \varphi_2$, for some $i > 0$, and $\mathfrak{M}, \pi[j] \models \varphi_1$, for every j with $0 < j < i$;
- $\mathfrak{M}, s \models \mathcal{A}(\varphi_1 \mathcal{U} \varphi_2) \Leftrightarrow$ for every π with $\pi[0] = s$, both $\mathfrak{M}, \pi[i] \models \varphi_2$, for some $i > 0$, and $\mathfrak{M}, \pi[j] \models \varphi_1$, for every j with $0 < j < i$.

The satisfaction clauses for the other formulas are similar to those for **QLTL** (with paths replaced by states). By analogy with **QLTL**, the other temporal operators can be defined as

$$\begin{aligned} \mathcal{E}\bigcirc\varphi &= \mathcal{E}(\perp \mathcal{U} \varphi); & \mathcal{A}\bigcirc\varphi &= \mathcal{A}(\perp \mathcal{U} \varphi); \\ \mathcal{E}\Diamond\varphi &= \varphi \vee \mathcal{E}(\top \mathcal{U} \varphi); & \mathcal{A}\Diamond\varphi &= \varphi \vee \mathcal{A}(\top \mathcal{U} \varphi); \\ \mathcal{E}\Box\varphi &= \neg\mathcal{E}\Diamond\neg\varphi; & \mathcal{A}\Box\varphi &= \neg\mathcal{A}\Diamond\neg\varphi. \end{aligned}$$

Now, if we replace in the formulas defined in Section 3 $\Diamond$ with $\mathcal{E}\Diamond$, $\Box$ with $\mathcal{A}\Box$, and $\bigcirc$ with $\mathcal{A}\bigcirc$, then we readily obtain results for **QCTL** analogous to those obtained in Section 3 for **QLTL**. This observation yields the following result:

Theorem 3. *Satisfiability for **QCTL** over predicate frames with both expanding and constant domains is Σ_1^1 -hard in languages with two individual variables and a single monadic predicate letter. Validity for **QCTL** over predicate frames with both expanding and constant domains is Π_1^1 -hard in languages with two individual variables and a single monadic predicate letter.*

The language of quantified alternating-time temporal logic **QATL** contains — in place of the two path quantifiers, $\mathcal{E}$ and $\mathcal{A}$, of **QCTL** — path quantifiers parametrised by groups of agents; the so obtained formulas are evaluated over concurrent game structures [Alur et al., 2002]. The **QCTL** path quantifiers $\mathcal{E}$ and $\mathcal{A}$ can, however, be expressed [Ibid.] as **QATL** path quantifiers parametrised by, respectively, the empty set of agents and the set of all the agents in the language. Such a replacement of path quantifiers gives us a recursive embedding of **QCTL** into **QATL** that affects neither the variables nor the predicate letters of formulas. Hence, from Theorem 3, we obtain the following result:

Theorem 4. *Satisfiability for **QATL** over predicate frames with both expanding and constant domains is Σ_1^1 -hard in languages with two individual variables and a single monadic predicate letter. Validity for **QATL** over predicate frames with both expanding and constant domains is Π_1^1 -hard in languages with two individual variables and a single monadic predicate letter.*

Obviously, Theorems 3 and 4 extend to the more expressive logics **QCTL**^{*} and **QATL**^{*}, since **QCTL** and **QATL** are fragments of, respectively, **QCTL**^{*} and **QATL**^{*} obtained by disallowing formulas of certain form.

5. Conclusion

We conclude with mentioning directions for future research.

First, an important open problem is identification of the minimal undecidable fragments of **QLTL** over finite domains (it is known that **QLTL** over finite domains behaves differently from **QLTL** over arbitrary domains [Hodkinson et al., 2000, Theorem 25]). The reduction used in the present paper essentially relied on the domains of the states to be infinite; therefore, the construction we used is not applicable to **QLTL** over finite domains. First-order logics over finite domains are usually computationally harder than logics over arbitrary domains [Trakhtenbrot, 1953; Libkin, 2004], even in languages with a small number of individual variables [Rybakov, Shkatov, 2019b] — is this true for **QLTL**? Monodic fragments of **QLTL** over finite domains are known to be decidable [Hodkinson et al., 2000, Theorem 26] — what are the minimal undecidable fragments?

Second, similar question arise for **QLTL** over structures with a finite set of states. Some undecidability results for this logic are known [Cerrito et al., 1999], but the gap between the decidable and the undecidable remains quite wide. We do know that, in structures with finite sets of states, we can model finite domains [Rybakov, Shkatov, 2020c; Rybakov, Shkatov, 2018c, Lemma 3.3]; while we would, in such a setting, be able to model all the predicate letters in

the language with a single monadic one, the number of variables needed would be quite large.

The last, and perhaps most technically challenging, direction for future research is the study of similar questions in the context of **QLTL** based on intuitionistic [Mints, 2000], rather than classical logic. Intuitionistic modal and temporal logics [Fischer Servi, 1977; Fischer Servi, 1980; Fischer Servi, 1984; Davoren, 2009], which formalise constructive modal and temporal reasoning, have been found of particular interest in computer science [Simpson, 1994; Davies, Pfenning, 2001; de Paiva et al., 2004; Yuse, Igarashi, 2006; Brewka et al., 2011; Davies, 2017]. Recently, motivated by applications in computer science, propositional linear-time temporal logics with an intuitionistic base have been proposed and studied [Maier, 2004; Boudou et al., 2017; Fernández-Duque, 2018; Diéguez et al., 2018; Boudou et al., 2019; Balbiani et al., 2019]. While a number of techniques for studying decidability and complexity of intuitionistic modal and temporal logics are known [Ono, 1977; Simpson, 1994; Grefe, 1998; Wolter, Zakharyashev, 1997; Wolter, Zakharyashev, 1999; Alechina, Shkatov, 2006; Gabbay et al., 2003, Chapter 10], no research, as far as we know, has been done on first-order temporal intuitionistic logic. If this direction is pursued, however, we expect to be able to draw on extensive research into computational properties of various fragments of both propositional [Nishimura, 1960; Rybakov, 2006; Rybakov, 2008] and first-order [Maslov et al., 1965; Mints, 1968; Gabbay, 1981; Kontchakov et al., 2005; Rybakov, Shkatov, 2019c; Rybakov, Shkatov, 2021b] intuitionistic logics.

Acknowledgements. The work on this paper, carried out at Tver State University during the second author’s visit, has been supported by Russian Science Foundation, project 21-18-00195.

References

- Abadi, 1989 – Abadi, M. “The power of temporal proofs”, *Theoretical Computer Science*, 1989, Vol. 65, No. 1, pp. 35–83.
- Abiteboul, 1996 – Abiteboul, S., Herr, L., Van den Bussche, J. “Temporal versus first-order logic to query temporal databases”, in: *Proceedings of the 15th International Conference on Principles of Databases (PODS ’96)*, 1996.
- Ajspur et al., 2013 – Ajspur, M., Goranko, V., Shkatov, D. “Tableau-based decision procedure for the multiagent epistemic logic with all coalitional operators for common and distributed knowledge”, *Logic Journal of the IGPL*, 2013, Vol. 21, No. 3, pp. 407–437.
- Alechina, Shkatov, 2006 – Alechina, N., Shkatov, D. “A general method for proving decidability of intuitionistic modal logics”, *Journal of Applied Logic*, 2006, Vol. 4, No. 3, pp. 219–230.

- Alur et al., 2002 – Alur, R., Henzinger, T.A., Kuperman, O. “Alternating-Time temporal logic”, *Journal of ACM*, 2002, Vol. 49, No. 5, pp. 672–713.
- Andréka et al., 1979 – Andréka, H., Németi, I., Sain, I. “Completeness problems in verification of programs and program schemes”, in: *Mathematical Foundations of Computer Science 1979. MFCS 1979*, ed. by J. Bečvář, Vol. 74 of *Lecture Notes in Computer Science*. Springer, 1979.
- Artale, Franconi, 2000 – Artale, A., Franconi, E. “A survey of temporal extensions of description logics”, *Annals of Mathematics and Artificial Intelligence*, 2000, Vol. 30, pp. 171–210.
- Artale et al., 2004 – Artale, A., Kontchakov, R., Kovtunova, A., Ryzhikov, V., Wolter, F., Zakharyashev, M. “First-order rewritability of ontology-mediated queries in linear temporal logic”, 2020. Preprint available at <https://arxiv.org/abs/2004.07221>
- Artemov, Dzhaparidze, 1990 – Artemov, S., Dzhaparidze, G. “Finite Kripke models and predicate logics of provability”, *The Journal of Symbolic Logic*, 1990, Vol. 55, No. 3, pp. 1090–1098.
- Baader et al., 2015 – Baader, F., Borgwardt, S., Lippmann, M. “Temporal query entailment in the description logic SHQ”, *Journal of Web Semantics*, 2015, Vol. 33, pp. 71–93.
- Baader et al., 2012 – Baader, F., Ghilardi, S., Lutz, C. “Ltl over description logic axioms”, *ACM Transactions on Computational Logic*, 2012, Vol. 13, No. 3, Article 21.
- Balbiani et al., 2019 – Balbiani, P., Boudou, J., Diéguez, M., Fernández-Duque, D. “Intuitionistic linear temporal logics”, 2019, *ACM Transactions on Computational Logic*, Vol. 21, Article 14.
- Börger et al., 1997 – Börger, E., Grädel, E., Gurevich, Y. *The Classical Decision Problem*. Springer, 1997.
- Borgwardt et al., 2015 – Borgwardt, S., Lippmann, M., Thost, V. “Temporalizing rewritable query languages over knowledge bases”, *Journal of Web Semantics*, 2015, Vol. 33, pp. 50–70.
- Boudou et al., 2017 – Boudou, J., Diéguez, M., Fernández-Duque D. “A decidable intuitionistic temporal logic”, in: *Proceedings of the 26th EACSL Annual Conference on Computer Science Logic (CSL’17)*, 2017, Vol. 82, pp. 14:1–14:17.
- Boudou et al., 2019 – Boudou, J., Diéguez, M., Fernández-Duque, D., Romero F. “Axiomatic systems and topological semantics for intuitionistic temporal logic”, in: F. Calimeri, N. Leone, M. Manna (eds.), *Logics in Artificial Intelligence. JELIA 2019*, Vol. 11468 of *Lecture Notes in Computer Science*. Springer, 2019, pp. 763–777.
- Bourgaux et al., 2019 – Bourgaux, C., Koopmann, P., Turhan, A.-Y. “Ontology-mediated query answering over temporal and inconsistent data”, *Semantic Web Journal*, 2019, Vol. 10, pp. 475–521.

- Braüner, Ghilardi, 2007 – Braüner, T., Ghilardi, S. “First-order modal logic”, in: P. Blackburn, J. Van Benthem, F. Wolter (eds.), *Handbook of Modal Logic*, Vol. 3 of *Studies in Logic and Practical Reasoning*. Elsevier, 2007, pp. 549–620.
- Brewka et al., 2011 – Brewka, G., Eiter, T., Truszczyński, M. “Answer set programming at a glance”, *Communications of the ACM*, 2011, Vol. 54, No. 12, pp. 92–103.
- Cerrito et al., 1999 – Cerrito, S., Mayer, M.C., Praud, S. “First order linear temporal logic over finite time structures”, in: H. Ganzinger, D. McAllester, A. Voronkov (eds.), *Logic for Programming and Automated Reasoning. LPAR 1999*, Vol. 1705 of *Lecture Notes in Computer Science*. Springer, 1999.
- Chagrov, Rybakov, 2003 – Chagrov, A., Rybakov, M. “How many variables does one need to prove PSPACE-hardness of modal logics?” in: P. Balbiani, N.-Y. Suzuki, F. Wolter, M. Zakharyashev (eds.), *Advances in Modal Logic*, Vol. 4. London: King’s College Publications, 2003, pp. 71–82.
- Chomicki, 1994 – Chomicki, J. “Temporal query languages: A survey”, in: D.M. Gabbay and H.J. Ohlbach (eds.), *Temporal Logic. ICTL 1994*, Vol. 827 of *Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence)*. Springer, 1994.
- Chomicki, Niwinski, 1993 – Chomicki, J., Niwinski, D. “On the feasibility of checking temporal integrity constraints”, *Journal of Computer and System Sciences*, 1995, Vol. 51, No. 3, pp. 523–535.
- Church, 1936 – Church, A. “A note on the ‘Entscheidungsproblem’”, *The Journal of Symbolic Logic*, 1936, Vol. 1, pp. 40–41.
- Clarke et al., 2000 – Clarke, E.M., Grumberg, O., Peled, D.A. *Model Checking*. Cambridge: MIT Press, 2000.
- Davies, Pfenning, 2001 – Davies, R., Pfenning, F. “A modal analysis of staged computation”, *Journal of the ACM*, 2001, Vol. 48, No. 3, pp. 555–604.
- Davies, 2017 – Davies, R. “A temporal logic approach to binding-time analysis”, *Journal of the ACM*, 2017, Vol. 64, No. 1, pp. 1–45.
- Davoren, 2009 – Davoren, J.M. “On intuitionistic modal and tense logics and their classical companion logics: Topological semantics and bisimulations”, *Annals of Pure and Applied Logic*, 2009, Vol. 161, No. 3, pp. 349–367.
- de Paiva et al., 2004 – de Paiva, V., Goré, R., Mendler, M. “Modalities in constructive logics and type theories”, *Journal of Logic and Computation*, 2004, Vol. 14, pp. 439–446.
- Demri et al., 2016 – Demri, S., Goranko, V., Lange, M. *Temporal Logics in Computer Science: Finite-state systems*, Vol. 58 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2016.
- Diéguez et al., 2018 – Diéguez, M., Fernández-Duque, D. “An intuitionistic axiomatization of ‘eventually’” in: G. Metcalfe, G. Bezhanishvili, G. D’Agostino and T. Studer (eds.), *Advances in Modal Logic*, Vol. 12, College Publications, 2018, pp. 199–218.
- Emerson, 1990 – Emerson, E.A. “Temporal and modal logic”, in: J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science*, 1990, Vol. B, pp. 995–1072.

- Emerson, Halpern, 1985 – Emerson, E.A., Halpern, J. “Decision procedures and expressiveness in temporal logic of branching time”, *Journal of Computer and System Sciences*, 1985, Vol. 30, No. 1, pp. 1–24.
- Enderton, 2011 – Enderton, H.B. *Computability Theory: An Introduction to Recursion Theory*. New York: Academic Press, 2011.
- Fagin et al., 1995 – Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y. *Reasoning About Knowledge*. Cambridge: MIT Press, 1995.
- Fernández-Duque, 2018 – Fernández-Duque, D. “The intuitionistic temporal logic of dynamical systems”, *Logical Methods in Computer Science*, 2018, Vol. 14, No. 3, pp. 1–35.
- Fischer, Ladner, 1979 – Fischer, M., Ladner, R.E. “Propositional dynamic logic of regular programs”, *Journal of Computer and System Sciences*, 1979, Vol. 18, pp. 194–211.
- Fischer Servi, 1977 – Fischer Servi, G. “On modal logic with an intuitionistic base”, *Studia Logica*, 1977, Vol. 36, No. 3, pp. 141–149.
- Fischer Servi, 1980 – Fischer Servi, G. “Semantics for a class of intuitionistic modal calculi”, in: M.L. Dalla Chiara (ed.), *Italian Studies in the Philosophy of Science*, pp. 59–72. Reidel; Dordrecht, 1980.
- Fischer Servi, 1984 – Fischer Servi, G. “Axiomatizations for some intuitionistic modal logics”, *Rendiconti del Seminario Matematico della Università di Padova*, 1984, Vol. 42, pp. 179–194.
- Fitting, Mendelsohn, 1998 – Fitting, M., Mendelsohn, R.L. *First-Order Modal Logic*, Vol. 277 of *Synthese Library*. Dordrecht: Kluwer Academic Publishers, 1998.
- Gabbay, 1981 – Gabbay, D. *Semantical Investigations in Heyting’s Intuitionistic Logic*. Netherlands: D. Reidel, 1981.
- Gabbay et al., 1994 – Gabbay, D., Hodkinson, I., Reynolds, M. *Temporal Logic: Mathematical Foundations and Computational Aspects, Volume 1*, Vol. 28 of *Oxford Logic Guides*. New York: Oxford University Press, 1994.
- Gabbay et al., 2003 – Gabbay, D., Kurucz, A., Wolter, F., Zakharyashev, M. *Many-Dimensional Modal Logics: Theory and Applications*, Vol. 148 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2003.
- Gabbay, Shehtman, 1993 – Gabbay, D., Shehtman, V. “Undecidability of modal and intermediate first-order logics with two individual variables”, *The Journal of Symbolic Logic*, 1993, Vol. 58, No. 3, pp. 800–823.
- Gabbay et al., 2009 – Gabbay, D., Shehtman, V., Skvortsov, D. *Quantification in Nonclassical Logic, Volume 1*, Vol. 153 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2009.
- Garson, 2001 – Garson, J.W. Quantification in modal logic. in: D.M. Gabbay and F. Guenther (eds.), *Handbook of Philosophical Logic*, Vol. 3, pp. 267–323. Springer; Dordrecht, 2001.
- Goldblatt, 1992 – Goldblatt, R. *Logics of Time and Computation*, Vol. 7 of *CSLI Lecture Notes*. Second edition. Stanford: Center for the Study of Language and Information, 1992.

- Goldblatt, 2011 – Goldblatt, R. *Quantifiers, Propositions and Identity: Admissible Semantics for Quantified Modal and Substructural Logics*. Lecture Notes in Logic. Cambridge: Cambridge University Press, 2011.
- Goranko, Shkatov, 2008 – Goranko, V., Shkatov, D. “Tableau-based decision procedures for the multi-agent epistemic logic with operators of common and distributed knowledge”, in: *Proceedings of the Sixth International Conference on Software Engineering and Formal Methods*, IEEE Press, 2008, pp. 237–246.
- Goranko, Shkatov, 2009a – Goranko, V., Shkatov, D. “Tableau-based decision procedure for full coalitional multiagent temporal-epistemic logic of linear time”, in: S. Artemov and A. Nerode (eds.), *International Symposium on Logical Foundations of Computer Science*, 2009, pp. 197–213.
- Goranko, Shkatov, 2009b – Goranko, V., Shkatov, D. “Tableau-based procedure for deciding satisfiability in the full coalitional multiagent epistemic logic”, in: C. Sierra, C. Castelfranchi, K.S. Decker, J. Sichman (eds.), *Proceedings of 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 09)*, 2009, pp. 969–976.
- Goranko, Shkatov, 2009c – Goranko, V., Shkatov, D. “Tableau-based decision procedure for the full coalitional multiagent temporal-epistemic logic of branching time”, in: *Proceedings of Formal Approaches to Multi-Agent Systems 2009*, 2009.
- Goranko, Shkatov, 2009d – Goranko, V., Shkatov, D. “Tableau-based decision procedures for logics of strategic ability in multiagent systems”, *ACM Transactions on Computational Logic*, 2009, Vol. 11, No. 1, pp. 3–51.
- Goranko, van Drimmelen, 2006 – Goranko, V., Shkatov, D. “Complete axiomatization and decidability of the alternating-time temporal logic”, *Theoretical Computer Science*, 2006, Vol. 353, No. 1–3, pp. 93–117.
- Grädel et al., 1997 – Grädel E., Kolaitis, P.G., Vardi, M.Y. “On the decision problem for two-variable first-order logic”, *Bulletin of Symbolic Logic*, 1997, Vol. 3, No. 1, pp. 53–69.
- Grefe, 1998 – Grefe, C. “Fischer Servi’s intuitionistic modal logic has the finite model property”, in: M. Kracht, M. de Rijke, H. Wansing, M. Zakharyashev (eds.), *Advances in Modal Logic*, Vol. 1, CSLI Publications, 1998, pp. 85–98.
- Halpern, 1995 – Halpern, J.Y. “The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic”, *Artificial Intelligence*, 1995, Vol. 75, No. 2, pp. 361–372.
- Halpern, Vardi, 1998 – Halpern, J.Y., Vardi, M.Y. “The complexity of reasoning about knowledge and time I: Lower bounds”, *Journal of Computer and System Sciences*, 1989, Vol. 38, No. 1, pp. 195–237.
- Harel, 1986 – Harel, D. “Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness”, *Journal of the ACM*, 1986, Vol. 33, pp. 224–248.
- Hodkinson et al., 2003 – Hodkinson, I., Kontchakov, R., Kurucz, A., Wolter, F., Zakharyashev, M. “On the computational complexity of decidable fragments of

- first-order linear temporal logics”, in: *10th International Symposium on Temporal Representation and Reasoning 2003 and Fourth International Conference on Temporal Logic. Proceedings*. IEEE, 2003, pp. 91–98.
- Hodkinson et al., 2000 – Hodkinson, I., Wolter, F., Zakharyashev, M. “Decidable fragments of first-order temporal logics”, *Annals of Pure and Applied Logic*, 2000, Vol. 106, pp. 85–134.
- Hodkinson et al., 2001 – Hodkinson, I., Wolter, F., Zakharyashev, M. “Monodic fragments of first-order temporal logics: 2000–2001 A.D.”, in: R. Nieuwenhuis and A. Voronkov (eds.), *Logic for Programming, Artificial Intelligence, and Reasoning. LPAR 2001*, Vol. 2250 of *Lecture Notes in Computer Science*. Springer, 2001, pp. 1–23.
- Hughes, Cresswell, 1996 – Hughes, G.E., Cresswell, M.J. *A New Introduction to Modal Logic*. Routledge, 1996.
- Huth, Ryan, 2004 – Huth, M., Ryan, M. *Logic in Computer Science: Modelling and Reasoning about Systems*, second edition. Cambridge: Cambridge University Press, 2004.
- Kontchakov et al., 2005 – Kontchakov, R., Kurucz, A., Zakharyashev, M. “Undecidability of first-order intuitionistic and modal logics with two variables”, *Bulletin of Symbolic Logic*, 2005, Vol. 11, No. 3, pp. 428–438.
- Kripke, 1962 – Kripke, S. “The undecidability of monadic modal quantification theory”, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 1962, Vol. 8, pp. 113–116.
- Libkin, 2004 – Libkin, L. *Elements of Finite Model Theory*. Springer, 2004.
- Maier, 2004 – Maier, P. “Intuitionistic LTL and a new characterization of safety and liveness”, in: J. Marcinkowski, A. Tarlecki (eds.), *Proceedings of the Computer Science Logic 18th International Workshop (CSL’04)*, Vol. 3210 of *Lecture Notes in Computer Science*, Springer, 2004, pp. 295–309.
- Manna, Pnueli, 1992 – Manna, Z., Pnueli, A. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer, 1992.
- Manna, Pnueli, 1995 – Manna, Z., Pnueli, A. *Temporal Verification of Reactive Systems: Safety*. Springer, 1995.
- Marx, 1999 – Marx, M. “Complexity of products of modal logics”, *Journal of Logic and Computation*, 1999, Vol. 9, No. 2, pp. 197–214.
- Maslov et al., 1965 – Maslov, S., Mints, G., Orevkov, V. “Unsolvability in the constructive predicate calculus of certain classes of formulas containing only monadic predicate variables”, *Soviet Mathematics Doklady*, 1965, Vol. 6, pp. 918–920.
- Merz, 1992 – Merz, S. “Decidability and incompleteness results for first-order temporal logics of linear time”, *Journal of Applied Non-Classical Logics*, 1992, Vol. 2, No. 2, pp. 139–156.
- Mints, 1968 – Mints, G. “Some calculi of modal logic”, *Trudy Matematicheskogo Instituta imeni V.A. Steklova*, 1968, Vol. 98, No. 88–111. (In Russian)
- Mints, 2000 – Mints, G. *A Short Introduction to Intuitionistic Logic*. New York: Kluwer Academic Publishers, 2000.

- Mortimer, 1975 – Mortimer, M. “On languages with two variables”, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 1975, pp. 135–140.
- Nishimura, 1960 – Nishimura, I. “On formulas of one variable in intuitionistic propositional calculus”, *The Journal of Symbolic Logic*, 1960, Vol. 25, No. 4, pp. 327–331.
- Ono, 1977 – Ono, H. “On some intuitionistic modal logics”, *Publications of the Research Institute for Mathematical Sciences*, 1977, Vol. 13, No. 3, pp. 687–722.
- Pnueli, 1986 – Pnueli, A. “Applications of temporal logic to the specification and verification of reactive systems: A survey of current trends”, in: J.W. de Bakker, W.P. de Roever, G. Rozenberg (eds.), *Current Trends in Concurrency*, Vol. 224 of *Lecture Notes in Computer Science*. Springer, 1986.
- Rogers, 1967 – Rogers, H. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, 1967.
- Rybakov, 2006 – Rybakov, M. “Complexity of intuitionistic and Visser’s basic and formal logics in finitely many variables”, in: G. Governatori, I.M. Hodkinson, Y. Venema (eds.), *Advances in Modal Logic 6*. Stanford: College Publications, 2006, pp. 393–411.
- Rybakov, 2008 – Rybakov, M. “Complexity of intuitionistic propositional logic and its fragments”, *Journal of Applied Non-Classical Logics*, 2008, Vol. 18, No. 2–3, pp. 267–292.
- Rybakov, Shkatov, 2018a – Rybakov, M., Shkatov, D. “Complexity and expressivity of propositional dynamic logics with finitely many variables”, *Logic Journal of the IGPL*, 2018, Vol. 26, No. 5, pp. 539–547.
- Rybakov, Shkatov, 2018b – Rybakov, M., Shkatov, D. “Complexity and expressivity of branching- and alternating-time temporal logics with finitely many variables”, in: B. Fischer, T. Uustalu (eds.), *Theoretical Aspects of Computing–ICTAC 2018*, Vol. 11187 of *Lecture Notes in Computer Science*. Springer, 2018, pp. 396–414.
- Rybakov, Shkatov, 2018c – Rybakov, M., Shkatov, D. “A recursively enumerable Kripke complete first-order logic not complete with respect to a first-order definable class of frames”, in: G. Metcalfe, G. Bezhanishvili, G. D’Agostino, T. Studer (eds.), *Advances in Modal Logic*, Vol. 12. Stanford: College Publications, 2018, pp. 531–540.
- Rybakov, Shkatov, 2019a – Rybakov, M., Shkatov, D. “Complexity of finite-variable fragments of propositional modal logics of symmetric frames”, *Logic Journal of the IGPL*, 2019, Vol. 27, No. 1, pp. 60–68.
- Rybakov, Shkatov, 2019b – Rybakov, M., Shkatov, D. “Trakhtenbrot theorem for classical languages with three individual variables”, in: *Proceedings of SAIC-SIT2019*. Article No. 19. ACM, 2019.
- Rybakov, Shkatov, 2019c – Rybakov, M., Shkatov, D. “Undecidability of first-order modal and intuitionistic logics with two variables and one monadic predicate letter”, *Studia Logica*, 2019, Vol. 107, No. 4, pp. 695–717.
- Rybakov, Shkatov, 2020a – Rybakov, M., Shkatov, D. “Recursive enumerability and elementary frame definability in predicate modal logic”, *Journal of Logic and Computation*, 2020, Vol. 30, No. 2, pp. 549–560.

- Rybakov, Shkatov, 2020b – Rybakov, M., Shkatov, D. “Algorithmic properties of first-order modal logics of the natural number line in restricted languages”, in: S. Negri, N. Olivetti, R. Verbrugge, G. Sandu (eds.), *Advances in Modal Logic*, Vol. 13. Stanford: College Publications, 2020, pp. 523–539.
- Rybakov, Shkatov, 2020c – Rybakov, M., Shkatov, D. “Algorithmic properties of first-order modal logics of finite Kripke frames in restricted languages”, *Journal of Logic and Computation*, 2020, Vol. 30, No. 7, pp. 1305–1329.
- Rybakov, Shkatov, 2021a – Rybakov, M., Shkatov, D. “Complexity of finite-variable fragments of products with **K**”, *Journal of Logic and Computation*, 2021, Vol. 31, No. 2, pp. 426–443.
- Rybakov, Shkatov, 2021b – Rybakov, M., Shkatov, D. “Algorithmic properties of first-order superintuitionistic logics of finite Kripke frames in restricted languages”, *Journal of Logic and Computation*, 2021, Vol. 31, No. 2, pp. 494–522.
- Rybakov, Shkatov, 2021c – Rybakov, M., Shkatov, D. “Algorithmic properties of first-order modal logics of linear Kripke frames in restricted languages”, *Journal of Logic and Computation*, 2021, Vol. 31, No. 5, pp. 1266–1288.
- Rybakov, Shkatov, 2021d – Rybakov, M., Shkatov, D. “Algorithmic properties of **QK4.3** and **QS4.3**”, *Smirnov Readings on Logic 2021*, 2021, pp. 50–54.
- Rybakov, Shkatov, 2021e – Rybakov, M., Shkatov, D. “Undecidability of the logic of partial quasiary predicates”. To appear in *Logic Journal of the IGPL*, doi 10.1093/jigpal/jzab018.
- Schewe, 2008 – Schewe, S. “ATL* satisfiability is 2EXPTIME-complete”, in: *Automata, Languages and Programming. ICALP 2008*, Vol. 5126 of *Lecture Notes in Computer Science*. Springer-Verlag Berlin Heidelberg, 2008, pp. 373–385.
- Shehtman, Shkatov, 2019 – Shehtman, V., Shkatov, D. “On one-variable fragments of modal predicate logics”, in: *Proceedings of SYSMICS2019*. Amsterdam: Institute for Logic, Language and Computation; University of Amsterdam, 2019, pp. 129–132.
- Shehtman, Shkatov, 2020 – Shehtman, V., Shkatov, D. “Some prospects for semi-products and products of modal logics”, in: *Short papers. AiML2020*. Helsinki: University of Helsinki, 2020, pp. 107–111.
- Simpson, 1994 – Simpson, A. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD thesis. Edinburgh: University of Edinburgh, 1995.
- Sistla, Clarke, 1985 – Sistla, A.P., Clarke, E.M. “The complexity of propositional linear temporal logics”, *Journal of ACM*, 1985, Vol. 32, No. 3, pp. 733–749.
- Spaan, 1993 – Spaan, E. *Complexity of Modal Logics*. PhD thesis. Amsterdam: University of Amsterdam, 1993.
- Szalas, 1986 – Szalas, A. “Concerning the semantic consequence relation in first-order temporal logic”, *Theoretical Computer Science*, 1086, Vol. 47, pp. 329–334.
- Szalas, Holenderski, 1988 – Szalas, A., Holenderski, L. “Incompleteness of first-order temporal logic with until”, *Theoretical Computer Science*, 1988, Vol. 57, pp. 317–325.

- Trakhtenbrot, 1953 – Trakhtenbrot, B.A. “On recursive separability”, *Doklady AN SSSR*, 1953, Vol. 88, pp. 953–956. (In Russian)
- Turing, 1936 – Turing, A.M. “On computable numbers, with an application to the ‘Entscheidungsproblem’”, *Proceedings of the London Mathematical Society, 2 series*, 1936/1937, Vol. 42, pp. 230–265.
- van der Hoek, Wooldridge, 2003 – van der Hoek, W., Wooldridge, M. “Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications”, *Studia Logica*, 2003, Vol. 75, No. 1, pp. 125–157.
- Vardi, Stockmeyer, 1985 – Vardi, M.Y., Stockmeyer, L. “Improved upper and lower bounds for modal logics of programs”, in: *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, 1985, pp. 240–251.
- Walther et al., 2006 – Walther, D., Lutz, C., Wolter, F., Wooldridge, M. “ATL satisfiability is indeed ExpTime-complete”, *Journal of Logic and Computation*, 2006, Vol. 16, No. 6, pp. 765–787.
- Wolter, Zakharyashev, 1997 – Wolter, F., Zakharyashev, M. “On the relation between intuitionistic and classical modal logics”, *Algebra and Logic*, 1997, Vol. 36, pp. 121–155.
- Wolter, Zakharyashev, 1999 – Wolter, F., Zakharyashev, M. “Intuitionistic modal logics as fragments of classical bimodal logics”, in: E. Orłowska (ed.), *Logic at Work*. Springer; Berlin, 1999, pp. 168–186.
- Wolter, Zakharyashev, 2000 – Wolter, F., Zakharyashev, M. “Temporalizing description logics”, in: D. Gabbay and M. de Rijke (eds.), *Frontiers of Combining Systems II*. London: Studies Press/Wiley, 2000, pp. 379–401.
- Wolter, Zakharyashev, 2001 – Wolter, F., Zakharyashev, M. “Decidable fragments of first-order modal logics” *The Journal of Symbolic Logic*, 2001, Vol. 66, pp. 1415–1438.
- Wolter, Zakharyashev, 2002 – Wolter, F., Zakharyashev, M. “Axiomatizing the monodic fragment of first-order temporal logic”, *Annals of Pure and Applied Logic*, 2002, Vol. 118, No. 1–2, pp. 133–145.
- Yuse, Igarashi, 2006 – Yuse, Y., Igarashi, A. “A modal type system for multi-level generating extensions with persistent code”, in: *Proceedings of the 8th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming (PPDP’06)*. New York, 2006, pp. 201–212.