A Bounded Domain Property for an Expressive Fragment of First-Order Linear Temporal Logic

Quentin Peyras
ONERA DTIS, Toulouse, France
Université fédérale de Toulouse, France
quentin.peyras@onera.fr

Julien Brunel
ONERA DTIS, Toulouse, France
Université fédérale de Toulouse, France
https://www.onera.fr/fr/staff/julien-brunel
julien.brunel@onera.fr

David Chemouil
ONERA DTIS, Toulouse, France
Université fédérale de Toulouse, France
https://www.onera.fr/fr/staff/david-chemouil
david.chemouil@onera.fr

Abstract

First-Order Linear Temporal Logic (FOLTL) is well-suited to specify infinite-state systems. However, FOLTL satisfiability is not even semi-decidable, thus preventing automated verification. To address this, a possible track is to constrain specifications to a decidable fragment of FOLTL, but known fragments are too restricted to be usable in practice. In this paper, we exhibit various fragments of increasing scope that provide a pertinent basis for abstract specification of infinite-state systems. We show that these fragments enjoy the Bounded Domain Property (any satisfiable FOLTL formula has a model with a finite, *bounded* FO domain), which provides a basis for complete, automated verification by reduction to LTL satisfiability. Finally, we present a simple case study illustrating the applicability and limitations of our results.

2012 ACM Subject Classification Theory of computation → Logic

Keywords and phrases First-Order Linear Temporal Logic, Bounded Domain Property, Finite Domain Property, Decidability

Digital Object Identifier 10.4230/LIPIcs.TIME.2019.15

Funding Work partly financed by the European Regional Development Fund (ERDF) through the Operational Programme for Competitiveness and Internationalisation (COMPETE2020) and by National Funds through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT) within project POCI-01-0145-FEDER-016826.

1 Introduction

First-Order Logic (FO) has proven to be useful to reason about the structure of a system, *i.e.*, the objects of the domain (which may be infinite), their relations and the properties they satisfy. Temporal logics, on the other hand, provide a natural way to specify the evolution of a system. First-Order Temporal Logics combine both dimensions and offer a flexible way of specifying systems with a rich structure, dynamic aspects and a possibly infinite number of states. First-Order Linear Temporal Logic (FOLTL) [7, 4] is the most studied among those.

However, formally verifying these properties is hard since FOLTL is not even semi-decidable. As we aim at verifying abstract specifications of infinite-state systems, a source of inspiration for the syntactic shape of fragments can be found in formal specification

approaches such as Lamport's TLA$^+$ [9] or the present authors' Electrum [2, 10]. In the latter for instance, a specification typically has the following form (ignoring relational and data structuring features):

$$\mathit{spec} = \mathit{init} \wedge \mathbf{G}\,\mathit{trans} \wedge \mathit{fair} \rightarrow \mathit{prop}$$

where:

- *init* is an FO formula that expresses *initial conditions* of the system;
- *trans* is an FOLTL formula that describes the *system transitions* and that only includes the LTL connective **X** (next) and first-order quantifiers;
- *fair* is an FOLTL formula, which expresses *fairness conditions* and thus includes nested LTL connectives **G** (always) and **F** (eventually);
- *prop* is an FOLTL formula that expresses a *property* expected of the system under specification. It is in principle arbitrarily complex but, in practice, for a large class of systems, it often remains in a relatively simple fragment of FOLTL.

Checking the validity of *spec* ($\models \mathit{spec}$) can be reduced to verifying that $\neg\mathit{spec}$ is *unsatisfiable* ($\mathrm{UNSAT}_{\mathrm{FOLTL}}(\neg\mathit{spec})$), with $\neg\mathit{spec} = \mathit{init} \wedge \mathbf{G}\,\mathit{trans} \wedge \mathit{fair} \wedge \neg\mathit{prop}$. Typically, however, $\neg\mathit{spec}$ does not belong to any formerly known decidable fragment of FOLTL.

Our main contribution is precisely to devise some novel *decidable* fragments of FOLTL encompassing formulas of the shape $\neg\mathit{spec}$.

We introduce in particular the Geneva[1] fragment, which consists of the set of NNF formulas of shape $\psi \wedge \mathbf{G}(\phi)$, where ψ is an FOLTL formula featuring existential quantifiers only at the root and where universal sub-formulas do not contain temporal connectives; and where ϕ is an FOLTL formula featuring only **X** and **F** as temporal connectives and where universal sub-formulas do not contain temporal connectives.

In practice, we prove that this fragment and some variants enjoy the *Bounded Domain Property* (BDP): any satisfiable formula (in any of these fragments) admits a model with finite, *bounded FO domain* (where the bound depends on the shape of the formula)[2]. Remark that the bound does not apply to the temporal dimension of models, only to the FO domain.

Written in a contrapositive way, provided $\neg\mathit{spec}$ belongs to one of these fragments, there is a bound k such that $\mathrm{UNSAT}_{\mathrm{FOLTL}}(\neg\mathit{spec})$ can be reduced to $\mathrm{UNSAT}^{\leqslant k}_{\mathrm{FOLTL}}(\neg\mathit{spec})$, where $\mathrm{UNSAT}^{\leqslant k}_{\mathrm{FOLTL}}$ means unsatisfiability in interpretation structures with FO domain of size $\leqslant k$.

Using this bound k, the FOLTL formula *spec* can be expanded into a plain LTL formula spec' (by unfolding quantifiers over the bounded domain). This way, the $\mathrm{UNSAT}^{\leqslant k}_{\mathrm{FOLTL}}(\neg\mathit{spec})$ problem is itself reduced to an $\mathrm{UNSAT}_{\mathrm{LTL}}(\neg\mathit{spec}')$. As LTL satisfiability is decidable, this ultimately yields a complete, automated decision procedure for the original problem.

Additionally, we make the following two remarks:

- for several of our fragments, the bound is linear in the size of formulas and exponential in certain formula-related criteria that are usually small in practice;
- for several fragments, the characterized bound is effectively reached, in the sense that $\mathrm{UNSAT}_{\mathrm{FOLTL}}(\neg\mathit{spec})$ can even be reduced to $\mathrm{UNSAT}^{=k}_{\mathrm{FOLTL}}(\neg\mathit{spec})$, which can in practice be leveraged to produce a smaller LTL formula to check for unsatisfiability.

The remainder of the article is organized as follows. In Sect. 2, we provide preliminary definitions about FOLTL. In Sect. 3, we exhibit axioms of infinity, i.e. formulas that *do not* enjoy the FDP, in order to guide the search for logical fragments enjoying the BDP. Then

[1] Geneva is a mnemonic for "**G**, Exists, Next/Eventually, (for)All".

[2] This work extends [8] where various simple fragments of FOLTL were studied. In particular, only one fragment including all LTL connectives, an extension of the classic Ramsey FO fragment (cf. Ex. 9), had been shown to enjoy the FDP.

we state some lemmas useful for subsequent proofs. In Sect. 4, we provide a step-by-step definition of a fragment of FOLTL that is relevant in the context of system specification. We establish that it enjoys the FDP and exhibit a bound on the FO domain. Then we illustrate our method on a toy example in Sect 5. Finally, we draw a comparison with other work in Sect. 6.

2 Syntax and Semantics of FOLTL

2.1 FOLTL

The basic vocabulary of FOLTL is defined out of a signature $\Sigma = (\mathcal{F}, \mathcal{R})$ where $\mathcal{F} = (\mathcal{F}_i)_{i\in\mathbb{N}}$ (resp. $\mathcal{R} = (\mathcal{R}_i)_{i\in\mathbb{N}}$) is a family of sets of *function* (resp. *predicate*) *symbols*), with $\mathcal{F}_i$ (resp. $\mathcal{R}_i$) the set of function (resp. predicate) symbols of *arity* i. We write *Const* for the set $\mathcal{F}_0$ of constant symbols. Given a set $\mathcal{V}$ of *variables*, the set $\mathcal{T}_{\Sigma,\mathcal{V}}$ of terms over Σ and $\mathcal{V}$ is defined in the usual way. Terms in $\mathcal{T}_{\Sigma,\varnothing}$ are called *closed terms*.

▶ **Definition 1** (Formulas). *Given a signature $\Sigma = (\mathcal{F}, \mathcal{R})$ and a set of variables $\mathcal{V}$, FOLTL* formulas *over Σ and $\mathcal{V}$ are defined inductively by the following grammar (with $x \in \mathcal{V}$, $r \in \mathcal{R}_n$ and every t_i in $\mathcal{T}_{\Sigma,\mathcal{V}}$):*

$$\psi ::= r(t_1, \ldots, t_n) \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\,\psi \mid \psi\,\mathbf{U}\,\psi \mid \forall x \cdot \psi \mid \exists x \cdot \psi$$

$\mathbf{X}$ and $\mathbf{U}$ stand for the "next" and "until" connectives. We also extend the set of temporal connectives by defining "eventually" as $\mathbf{F}\,\psi = \top\,\mathbf{U}\,\psi$, "always" as $\mathbf{G}\,\psi = \neg\,\mathbf{F}(\neg\psi)$ and "releases" as $\psi_1\,\mathbf{R}\,\psi_2 = \neg((\neg\psi_1)\,\mathbf{U}\,(\neg\psi_2))$. Similarly, classical propositional connectives $\wedge$, $\Rightarrow$ and $\Leftrightarrow$ are defined in the natural way.

Additionally,

- we write $\psi[x]$ for a formula ψ having x as a free variable.
- We write $\mathrm{FV}(\phi)$ for the set of *free variables* of a formula, defined in the obvious way. Also, a formula ϕ is said to be *closed* if $\mathrm{FV}(\phi) = \varnothing$.
- Given a formula ψ, we write $\mathcal{T}_\psi$ for the set of terms, including sub-terms, appearing in ψ.
- Classically, a formula is in *negation normal form* (NNF) if negations only appear in front of predicate symbols.
- We denote by $\mathrm{LTL}_{\Sigma,\mathcal{V}}$ the set of FOLTL formulas, built over Σ with variables in $\mathcal{V}$, that do not contain any first-order quantifier. We write $\mathrm{LTL}_{\Sigma,\mathcal{V}}(\mathbf{X})$ (resp. $\mathrm{LTL}_{\Sigma,\mathcal{V}}(\mathbf{X},\mathbf{F})$) for the set of formulas from $\mathrm{LTL}_{\Sigma,\mathcal{V}}$ that are in NNF and that contain no other temporal connective than $\mathbf{X}$ (resp. $\mathbf{X}$ and $\mathbf{F}$).
- A formula l is called literal if $l = r(t_1, \ldots, t_n)$ or $l = \neg r(t_1, \ldots, t_n)$ where $x \in \mathcal{V}$, $r \in \mathcal{R}_n$ and every t_i in $\mathcal{T}_{\Sigma,\mathcal{V}}$).

We now introduce the semantics of FOLTL. In the interpretation structures defined below, the interpretation of predicates *varies* over time while that of function symbols *does not*. Notice we rely on the Kleene star in the definition.

▶ **Definition 2** ((Interpretation) Structure). *Given a signature $\Sigma = (\mathcal{F}, \mathcal{R})$, an* (interpretation) structure $\mathcal{M}$ (over Σ) *is a triple (D, σ, ρ) where:*

- *D, called the domain, is a non-empty set.*
- *σ is a map s.t. for any $c \in \mathcal{F}_0$, $\sigma(c) \in D$, and for any $f \in \mathcal{F}_n$, $\sigma(f) : D^n \to D$.*
- *$\rho : \mathbb{N} \times D^\star \to \mathcal{P}(\mathcal{R})$ is a map s.t. for any instant $i \in \mathbb{N}$ and any $\vec{a} = (a_1 \ldots, a_n) \in D^\star$, $\rho_i(\vec{a}) \subseteq \mathcal{R}_n$.*

$\mathcal{M}$ is said to be domain-finite *if D is finite. We also define the* domain size *(simply called* size *in the remainder of this paper) of $\mathcal{M}$ as $|D|$.*

▶ Remark 3 (Type of ρ). Usually, FOLTL structures would be defined with ρ a function $\mathbb{N} \times \mathcal{R} \to \mathcal{P}(D^\star)$ mapping at any instant a predicate to the set of tuples (of the domain) satisfying it. We turn this definition upside down, which is trivially equivalent to the classical one, to simplify the presentation of forthcoming definitions (in particular, partial structures introduced in Def. 10) and proofs.

▶ **Definition 4** (Assignment). *An* assignment $\mathcal{C}$ *in a domain* D *for variables in* $\mathcal{V}$ *is a map* $\mathcal{V} \to D$. *We write* $\mathcal{C}[x \mapsto d]$ *the assignment defined as* $\mathcal{C}[x \mapsto d](x) = d$ *and* $\mathcal{C}[x \mapsto d](y) = \mathcal{C}(y)$ *if* $y \neq x$. *The* extension *of* $\mathcal{C}$ *to* terms*, also written* $\mathcal{C}$*, is defined in the obvious way.*

▶ **Definition 5** (Satisfaction). *Given a structure* $\mathcal{M} = (D, \sigma, \rho)$ *and an assignment* $\mathcal{C}$*, the* satisfaction *relation* $\vDash$ *is defined by induction on formulas, for any* $i \in \mathbb{N}$*, as follows:*

- $\mathcal{M}, i, \mathcal{C} \vDash r(t_1, \ldots, t_n)$ *iff* $r \in \rho_i(\mathcal{C}(t_1), \ldots, \mathcal{C}(t_n))$;
- $\mathcal{M}, i, \mathcal{C} \vDash \neg\phi$ *iff* $\mathcal{M}, i, \mathcal{C} \nvDash \phi$;
- $\mathcal{M}, i, \mathcal{C} \vDash \phi_1 \vee \phi_2$ *iff* $\mathcal{M}, i, \mathcal{C} \vDash \phi_1$ *or* $\mathcal{M}, i, \mathcal{C} \vDash \phi_2$;
- $\mathcal{M}, i, \mathcal{C} \vDash \mathbf{X}\,\phi$ *iff* $\mathcal{M}, i+1, \mathcal{C} \vDash \phi$;
- $\mathcal{M}, i, \mathcal{C} \vDash \phi_1\,\mathbf{U}\,\phi_2$ *iff there exists* $k \in \mathbb{N}$ *s.t.* $\mathcal{M}, i+k, \mathcal{C} \vDash \phi_2$ *and for every* $0 \leq j < k$*, we have* $\mathcal{M}, i+j, \mathcal{C} \vDash \phi_1$;
- $\mathcal{M}, i, \mathcal{C} \vDash \exists y \cdot \phi$ *iff there exists* $d \in D$ *s.t.* $\mathcal{M}, i, \mathcal{C}[y \mapsto d] \vDash \phi$;
- $\mathcal{M}, i, \mathcal{C} \vDash \forall x \cdot \phi$ *iff for every* $d \in D$*, we have* $\mathcal{M}, i, \mathcal{C}[x \mapsto d] \vDash \phi$.

Given a closed formula ϕ*, we write* $\mathcal{M}, k \vDash \phi$ *if* $\mathcal{M}, k, [] \vDash \phi$*, where* $[]$ *is the empty assignment.*

Let ϕ, ϕ' be two FOLTL formulas. If for any structure $\mathcal{M}$ and an assignment $\mathcal{C}$, we have $\mathcal{M}, 0, \mathcal{C} \vDash \phi$ iff $\mathcal{M}, 0, \mathcal{C} \vDash \phi'$ then we say that ϕ and ϕ' are logically equivalent, written $\phi \equiv \phi'$.

▶ **Definition 6** (Finite Domain Property, Bounded Domain Property). *A closed formula* ϕ *of FOLTL enjoys the* finite domain property *(FDP) if* ϕ *is not satisfiable, or there is a domain-finite structure* $\mathcal{M}$ *s.t.* $\mathcal{M}, 0 \vDash \phi$. *Additionally, if the bound is computable, the formula is said to enjoy the Bounded Domain Property (BDP). A fragment of a logic enjoys the FDP (resp. BDP) if every formula in this fragment does.*

▶ Remark 7 (BDP and decidability). For pure FO, if a fragment enjoys the FDP (usually called the *Finite Model Property*), then it is decidable. As FOLTL is not recursively enumerable (contrary to FO), the FDP does not suffice to show decidability of a given fragment, while the BDP does.

▶ Remark 8 (BDP and complexity). If a fragment enjoys the BDP, then from the expression of the bound on the domain, we can easily deduce an upper bound of the complexity of satisfiability for this fragment, using the results from [8]. Indeed, in [8], the complexity of the satisfiability problem on bounded models is studied for full FOLTL.

▶ **Example 9.** The following fragments of FO enjoy the FDP (following the book and notations of Börger et al. [3]):

- $[\exists^\star\forall^\star, all]_=$(Ramsey 1930) the class of formulas with quantifier prefix $\exists^\star\forall^\star$, without function symbols, with arbitrary predicate symbols, with equality.
- $[\exists^\star, all, all]_=$(Gurevich 1976) the class of formulas with quantifier prefix $\exists^\star$, with arbitrary predicate and function symbols, with equality.

2.2 Partial Structures

We defined the notion of structures for FOLTL, however proving the BDP requires to define a model in several steps. Indeed, we will need to define predicate interpretations for a finite numbers of instants, and to define the truth values of predicates for the remaining time later

on. This is clumsy to do with structures since we will need to redefine the entire structure at each step. For this reason, we introduce a notion of partial structures, which is easier to handle.

▶ **Definition 10** (Partial (interpretation) structures). *A* partial (interpretation) structure $\mathcal{M}$ *(over* Σ*) is a triple* (D, σ, ρ) *satisfying the same conditions as in Def.* 2 *except that* ρ *is a partial function. We denote by* $\rho_i(\vec{x}) = \bot$ *the fact that* ρ *is not defined on the pair* $(i, \vec{x})$*.*

Structures are then the maximal elements of the set of partial structures for the following partial order.

▶ **Definition 11** (Extension ordering of partial structures). *Given two partial structures* $\mathcal{M} = (D, \sigma, \rho)$ *and* $\mathcal{M}' = (D', \sigma', \rho')$*, we define the partial order* $\preccurlyeq$ *over partial structures as follows:* $\mathcal{M}'$ extends $\mathcal{M}$*, written* $\mathcal{M} \preccurlyeq \mathcal{M}'$*, iff* $D = D'$*,* $\sigma = \sigma'$*, and* $\rho_i(\vec{a}) \neq \bot$ *implies* $\rho'_i(\vec{a}) = \rho_i(\vec{a})$*.*

This allows a natural generalization of satisfaction to partial structures by saying that a partial structure satisfies a formula if *all* its extensions that are structures satisfy it.

▶ **Definition 12** (Semantics over partial structures I). *Given a partial structure* $\mathcal{M}$*, we say that* $\mathcal{M}, i, \mathcal{C} \vDash \phi$ *iff for all structure* $\mathcal{M}'$ *s.t.* $\mathcal{M} \preccurlyeq \mathcal{M}'$*, we have* $\mathcal{M}', i, \mathcal{C} \vDash \phi$*.*

There is another, natural way to define the semantics over partial structures. We introduce it as it will be required in forthcoming proofs. This semantics can be defined by induction on formulas in NNF. Such a restriction is necessary because we cannot evaluate the truth value of $\neg\phi$ out of that of ϕ, therefore we cannot define a general semantics for the "not" connective. This is because if a partial structure can be extended to either satisfy ϕ or satisfy $\neg\phi$, then this partial structure satisfies neither of these formulas.

▶ **Definition 13** (Semantics over partial structures II). *Given a partial structure* $\mathcal{M} = (D, \sigma, \rho)$ *and an assignment* $\mathcal{C}$*, the* satisfaction *relation* $\Vdash$ *is defined by induction on formulas in negation normal form (NNF), for all non-negative integers* i *as follows:*

- $\mathcal{M}, i, \mathcal{C} \Vdash r(t_1, \ldots, t_n)$ *iff* $r \in \rho_i(\mathcal{C}(t_1), \ldots, \mathcal{C}(t_n))$*.*
- $\mathcal{M}, i, \mathcal{C} \Vdash \neg r(t_1, \ldots, t_n)$ *iff* $\rho_i(\mathcal{C}(t_1), \ldots, \mathcal{C}(t_n)) \neq \bot$ *and* $r \notin \rho_i(\mathcal{C}(t_1), \ldots, \mathcal{C}(t_n))$*.*
- $\mathcal{M}, i, \mathcal{C} \Vdash \phi_1 \wedge \phi_2$ *if and only if* $\mathcal{M}, i, \mathcal{C} \Vdash \phi_1$ *and* $\mathcal{M}, i, \mathcal{C} \Vdash \phi_2$*.*
- $\mathcal{M}, i, \mathcal{C} \Vdash \phi_1 \vee \phi_2$ *if and only if* $\mathcal{M}, i, \mathcal{C} \Vdash \phi_1$ *or* $\mathcal{M}, i, \mathcal{C} \Vdash \phi_2$*.*
- $\mathcal{M}, i, \mathcal{C} \Vdash \mathbf{X}\,\phi$ *iff* $\mathcal{M}, i+1, \mathcal{C} \Vdash \phi$*.*
- $\mathcal{M}, i, \mathcal{C} \Vdash \phi_1\,\mathbf{U}\,\phi_2$ *iff there exists* $k \in \mathbb{N}$ *s.t.* $\mathcal{M}, i+k, \mathcal{C} \vDash \phi_2$ *and for every integer* $0 \leq j < k$*, we have* $\mathcal{M}, i+j, \mathcal{C} \Vdash \phi_1$*.*
- $\mathcal{M}, i, \mathcal{C} \Vdash \phi_1\,\mathbf{R}\,\phi_2$ *iff for each* $k \in \mathbb{N}$ $\mathcal{M}, i+k, \mathcal{C} \Vdash \phi_2$ *or there exists an integer* j *s.t.* $\mathcal{M}, i+j, \mathcal{C} \Vdash \phi_1$ *and for every integer* $0 \leq k \leq j$*,* $\mathcal{M}, i+k, \mathcal{C} \Vdash \phi_2$*.*
- $\mathcal{M}, i, \mathcal{C} \Vdash \exists y \cdot \phi(y)$ *if and only if there exists* $d \in D$ *s.t.* $\mathcal{M}, i, \mathcal{C}[y \mapsto d] \Vdash \phi(y)$*.*
- $\mathcal{M}, i, \mathcal{C} \Vdash \forall x \cdot \phi(x)$ *if and only if for every* $d \in D$*, we have* $\mathcal{M}, i, \mathcal{C}[x \mapsto d] \Vdash \phi(x)$*.*

▶ **Lemma 14** (Equivalence of semantics). *Given a partial structure* $\mathcal{M}$*, a formula* ϕ *in NNF,* $k \in \mathbb{N}$ *and an assignment* $\mathcal{C}$*, we have* $\mathcal{M}, k, \mathcal{C} \vDash \phi$ *iff* $\mathcal{M}, k, \mathcal{C} \Vdash \phi$*.*

▶ **Definition 15** (Enrichment of a structure). *Given a partial structure* $\mathcal{M} = (D, \sigma, \rho)$ *s.t.* $\rho_i(\vec{d}) = \bot$*, we define the* enrichment *of* $\mathcal{M}$ *at instant* i *on tuple* $\vec{d}$ *for* $A \in \mathcal{P}(\mathcal{R})$*, written* $\mathcal{M}[(i, \vec{d}) \mapsto A]$*, as the triple* (D, σ, ρ') *where:* $\rho'_i(\vec{d}) = A$ *and for any* $j \in \mathbb{N}$ *and any tuple* $\vec{d'}$*,* $\rho'_j(\vec{d'}) = \rho_j(\vec{d'})$ *if* $(j, \vec{d'}) \neq (i, \vec{d})$*. Notice that* $\mathcal{M}[(i, \vec{d}) \mapsto A]$ *is an extension of* $\mathcal{M}$*.*

Some sort of induction is required to extend a partial structure for tuples over all instants of time. It is possible to proceed by extending a partial structure step-by-step using the previous definition. The result is then an increasing sequence of partial structures. Intuitively, it can be seen that such a sequence somehow converges to a partial structure where all steps of extension have been performed on it. The following definition formalizes this notion.

▶ **Definition 16** (Limit structure). *Let $(\mathcal{M}^k)_{k\in\mathbb{N}}$ be a $\preccurlyeq$-increasing sequence of partial structures, with $\mathcal{M}^k = (D, \sigma, \rho^k)$. Then we define the (partial)* limit structure *$\mathcal{M}^\infty = (D, \sigma, \rho^\infty)$ s.t., for any $i \in \mathbb{N}$ and vector $\vec{d} \in D^\star$: (1) if there exists k s.t. $\rho_i^k(\vec{d}) \neq \bot$, then $\rho_i^\infty(\vec{d}) = \rho_i^k(\vec{d})$; (2) if for every $k \in \mathbb{N}$ we have $\rho_i^k(\vec{d}) = \bot$, then $\rho_i^\infty(\vec{d}) = \bot$.*

As we focus on the BDP, we aim at building a domain-finite model of a formula out of any structure satisfying it. However, to ensure that we have a general method working for a fragment as expressive as possible, we need to make this domain-finite model as similar as possible to the original one. For this reason, we define the following notion of partial embedding. Informally, this embedding between two partial structures expresses that any element of the domain of the former has, for each instant, an equivalent element in the domain of the latter, meaning they satisfy the same predicates. In the case of n-ary predicates, two tuples with one-to-one equivalent elements are considered equivalent, so they satisfy the same predicates. A partial embedding is used to deduce information about the satisfaction of non-temporal universally quantified properties at some particular instant.

▶ **Definition 17** (Partial embedding). *Let $\mathcal{M}_0 = (D_0, \sigma_0, \rho^0)$, $\mathcal{M}_1 = (D_1, \sigma_1, \rho^1)$ be two partial structures and $f : \mathbb{N} \times D_0 \nrightarrow D_1$ be a partial function. We say that f is a* (partial) embedding *from $\mathcal{M}_0$ to $\mathcal{M}_1$, denoted $\mathcal{M}_0 \overset{f}{\hookrightarrow} \mathcal{M}_1$, if there exists $m \in \mathbb{N}$ s.t.:*

- *for each $c \in Const$, each $i \in \mathbb{N}$, we have $f_i(\sigma_0(c)) = \sigma_1(c)$;*
- *for each $g \in \mathcal{F}_n$ $(n > 0)$, $\vec{d} \in D_0^n$, and each $i \in \mathbb{N}$ s.t. $f_i(\vec{d}) \neq \bot$[3], $f_i(\sigma_0(g)(\vec{d})) = \sigma_1(g)(f_i(\vec{d}))$; and*
- *for each $\vec{d} \in D_0^\star$ and each $i \in \mathbb{N}$, if $f_{i+m}(\vec{d}) \neq \bot$ then $\rho_i^0(\vec{d}) = \rho_i^1(f_{i+m}(\vec{d}))$, otherwise $\rho_i^0(\vec{d}) = \bot$.*

3 Preliminary Results

Here we exhibit various formulas of FOLTL that *do not* enjoy the FDP, which hints on possible directions to find a fragment that *does* enjoy it. Then, we introduce some technical lemmas about elementary fragments of FOLTL, which will be useful to establish the BDP of the fragments studied in Sect. 4.

3.1 Axioms of Infinity

In this section, we report on various ways not to enjoy the FDP. Our study of FOLTL fragments was partly guided by the need to avoid syntactic fragments. We call *axiom of infinity* an FOLTL formula that does not satisfy the FDP. Finding such axioms is easy, even with strong constraints on first-order quantifiers.

Due to some results from [8], we start our study with formulas featuring existential quantifiers. For instance, the following axiom of infinity involves only one existential quantifier: $\mathbf{G}(\exists y \cdot P(y) \wedge \mathbf{X}\,\mathbf{G}\,\neg P(y))$. Indeed, to satisfy this formula, we need to find some element in the domain satisfying P at each instant of time; however this element will never satisfy P again so an infinite domain is needed to pick a different element at every instant.

[3] $f_i(\vec{d}) \neq \bot$ denotes the fact that $f_i(\vec{d})$ is defined

It then appears that existential quantification under a $\mathbf{G}$ connective can be problematic. However, this problem occurs only when several *nested* $\mathbf{G}$ connectives appear in the formula. Notice that nesting $\mathbf{G}$ connectives is often unnecessary in practical system specifications.

Therefore let us now focus on cases where we have a formula of the form $\mathbf{G}(\exists y \cdot \psi[y])$ where ψ does not contain any $\mathbf{G}$ or first-order quantifier.

Now, what about universal quantification? Unfortunately, even with a prefix within the Ramsey fragment, axioms of infinity can be found, such as the following: $\mathbf{G}(\exists y \forall x \cdot \neg P(y) \wedge \mathbf{X}\, P(y) \wedge (P(x) \Rightarrow \mathbf{X}\, P(x)))$. Here, the universal quantifier allows us to specify by induction that any element in the domain used for the existential quantifier satisfies $\mathbf{G}\, P(y)$. This is actually similar to the first axiom. In order to avoid this behaviour, a new restriction is needed. A possibility is to forbid the use of temporal connectives under the scope of a universal quantifier.

Another issue lies in the use of constant predicates (predicates whose value does not change along time). Assume we are given a constant order $<$ (axiomatized by universally quantified formula without temporal connectives). Then the following formula defines an axiom of infinity: $\mathbf{G}(\exists y \cdot P(y) \wedge \mathbf{X}(\forall x \cdot P(x) \Rightarrow y < x))$. Indeed it forces at each instant the existence of an element in the domain which is greater than all elements that have already been used. Satisfying the formula then requires to have an infinite domain.

Thus, to obtain a fragment enjoying the BDP, one should at least:

- forbid nested $\mathbf{G}$ connectives;
- forbid temporal connectives under the scope of a universal quantifier;
- and forbid constant predicates if universal quantifiers are allowed.

3.2 Preliminary Lemmas

We now introduce lemmas that are basic elements of the BDP proof for the FOLTL fragments studied in this article.

The following lemma allows us to consider a formula with a well-suited syntactic form for the upcoming proofs (*without* impact on computed bounds). Indeed, we will need to define interpretations of predicates such that a formula is true at every instant. However, in case of a disjunction, there may be various ways to satisfy a formula. For example, consider $\phi = (a \Rightarrow \mathbf{X}\, b) \wedge (a \Rightarrow \mathbf{F}\, c)$; in this case, at every instant, ϕ may be satisfied by having $\neg a$ or $\mathbf{X}\, b \wedge \mathbf{F}\, c$. So we transform ϕ into a disjunctive normal form allowing us to differentiate and pick in which way it can be satisfied. In the case of ϕ, we obtain $(\neg a) \vee (\mathbf{X}\, b \wedge \mathbf{F}\, c)$. Within each disjunct, we distinguish between the sub-formulas under an $\mathbf{F}$ connective (which need to be satisfied at an unspecified instant) and the other ones (which need to be satisfied at a specified number of instants, depending on the number of nested $\mathbf{X}$ connectives).

▶ **Lemma 18** (Disjunctive normal form (DNF)). *If ϕ is a formula in $LTL_{\Sigma,\mathcal{V}}(\mathbf{X},\mathbf{F})$ then there exists $\psi \equiv \phi$ s.t.: (1) ψ is a disjunction of the form $\psi_1 \vee \ldots \vee \psi_n$ (notice that each ψ_i is in NNF); (2) Each ψ_i is a conjunction of the form $\alpha_i \wedge \mathbf{F}\, \beta_{i,1} \wedge \ldots \wedge \mathbf{F}\, \beta_{i,j}$, with $\alpha_i = \mathbf{X}^{n_{i,1}}\, \ell_{i,1} \wedge \ldots \wedge \mathbf{X}^{n_{i,k_i}}\, \ell_{i,k_i}$ (writing $\mathbf{X}^n$ for a sequence of n $\mathbf{X}$ connectives) and where each $\ell_{i,k}$ is a literal and each $\beta_{i,k}$ is in $LTL_{\Sigma,\mathcal{V}}(\mathbf{X},\mathbf{F})$.*

▶ Remark 19 (Inocuity of the DNF transformation). The transformation of a formula into DNF induces an exponential blow-up. In this paper, it is only used to prove the BDP of the considered fragments: since the considered bounds are not affected by the transformation in DNF, the blow-up does not influence the complexity of the decision procedure.

The size of the finite model resulting from the construction presented in this paper depends on the depth of nested $\mathbf{X}$ connectives. For example, there is a structure of size 1 satisfying $\mathbf{G}(\exists y \cdot P(y))$. However any structure satisfying $\mathbf{G}(\exists y \cdot P(y) \wedge \mathbf{X}(\neg P(y)) \wedge \mathbf{X}\,\mathbf{X}(\neg P(y))$ is at least of size 3. This depends on the number of instants it refers to using $\mathbf{X}$ connectives:

▶ **Definition 20** (Stride of a formula). *Given a formula ϕ in DNF, we define its* stride K_ϕ *as the maximal depth of nested $\mathbf{X}$ connectives not under an $\mathbf{F}$, that is $K_\phi = \max_{i=1..n} \max_{j=1..k_i} n_{i,j}$ (with $n_{i,j}$ following the notations of Lemma 18).*

The following lemma applies to formulas containing only $\mathbf{X}$ and $\mathbf{F}$ connectives, as well as featuring only existential quantification over a single variable. Given such a formula, a model of this formula and a partial structure where constant symbols are interpreted as in the model of the formula, the lemma states that we can extend this partial structure into a partial model of the formula by providing an interpretation for the predicates (1) for a finite set of instants only and (2) over a single element in the domain.

▶ **Lemma 21.** *Consider a formula ψ in $LTL_{\Sigma,\{y\}}(\mathbf{X},\mathbf{F})$ and a structure $\mathcal{M} = (\mathcal{D}, \sigma, \rho)$ s.t. $\mathcal{M}, k \vDash \exists y \cdot \psi[y]$ for some $k \in \mathbb{N}$. Consider also a partial structure $\mathcal{M}_0 = (\mathcal{D}, \sigma_0, \rho^0)$ s.t. $\mathcal{M}_0 \overset{f^0}{\hookrightarrow} \mathcal{M}$ and s.t. there exists some a in $\mathcal{D}$ s.t. for each integer $j \geq k$ we have $f^0_j(a) = \bot$. Then, there exists an integer $k' > k$ (where $k' = k + K_\psi + 1$ if $\psi \in LTL_{\Sigma,\{y\}}(\mathbf{X})$) and a structure $\mathcal{M}_1 = (\mathcal{D}, \sigma_1, \rho^1)$ satisfying:*

- *$\mathcal{M}_1 \overset{f^1}{\hookrightarrow} \mathcal{M}$ for some f^1,*
- *for any $x \in \mathcal{D}$ and any $i \geq k'$, $f^1_i(x) \neq \bot$ iff $x \in \sigma_1(\mathcal{T}_{\Sigma,\varnothing})$,*
- *for any $i \in \mathbb{N}$ s.t. $k \leq i < k'$, and any $x \in D$, $x \neq a$ implies $f^0_i(x) = f^1_i(x)$,*
- *$\mathcal{M}_0 \preccurlyeq \mathcal{M}_1$,*
- *$\mathcal{M}_1, k, [y \mapsto a] \vDash \psi[y]$.*

Proof. First notice that the truth value of a formula in $\mathrm{LTL}_{\Sigma,\{y\}}(\mathbf{X},\mathbf{F})$ can be determined by only "looking at" a finite set of instants I, in the sense that changing the interpretation of predicates outside I does not change the truth value of the formula. This can be shown for instance by induction on the number of nested $\mathbf{F}$.

Let ψ be a formula in $\mathrm{LTL}_{\Sigma,\{y\}}(\mathbf{X},\mathbf{F})$ s.t. $\mathcal{M}, k \models \exists y \cdot \psi[y]$. Let k' be the greatest instant in the set I as introduced above. Let d be an element in the domain such that $\mathcal{M}, k, [y \mapsto d] \models \psi[y]$. Then, we can extend $\mathcal{M}_0$ into $\mathcal{M}_1$ in a way s.t. $f^1_i(a) = d$ and $\rho^1_i(a) = \rho_i(d)$ for $i \in [k, k']$. ◀

The next lemma focuses on formulas containing $\mathbf{X}$ connectives only. It establishes that formulas of the form $\mathbf{G}(\exists y \cdot \psi)$, where the only temporal connective in ψ is $\mathbf{X}$, enjoy the BDP. However, this lemma is formulated in a more suitable way for the proof of Theorem 26. In particular, we limit the result to a finite temporal window $[k_1, k_2]$.

▶ **Lemma 22.** *Assume that there exists $k_1, k_2 \in \mathbb{N}$ s.t. for any integer $i \in [k_1, k_2]$ we have $\mathcal{M}, i \vDash \exists y \cdot \alpha[y]$, where $\alpha \in LTL_{\Sigma,\{y\}}(\mathbf{X})$. Let $\mathcal{M}^0$ be a partial structure s.t. $\mathcal{M}^0 \overset{f^0}{\hookrightarrow} \mathcal{M}$ for some f^0, and there exists $A = \{a_0, \ldots, a_{K_\alpha}\}$ s.t. for each integer $j \in [k_1, k_2 + K_\alpha]$ and all $a \in A$, we have $f^0_j(a) = \bot$. Then there exists $\mathcal{M}^1$ s.t.:*

- *$\mathcal{M}^1 \overset{f_1}{\hookrightarrow} \mathcal{M}$ for some f^1;*
- *$\mathcal{M}^0 \preccurlyeq \mathcal{M}^1$;*
- *$f^1_j(x) \neq f^0_j(x)$ implies that $j \in [k_1, k_2 + K_\alpha]$ and $x \in A$;*
- *For any $i \in [k_1, k_2]$, there exists $m \leq K_\alpha$ s.t. $\mathcal{M}^1, i, [y \mapsto a_m] \vDash \alpha[y]$.*

Figure 1 First step of the partial structure construction.

	0	1	2	3	...
a_0	P	$\neg P$	?	?	...
a_1	?	P	$\neg P$	?	...
a_2	?	?	P	$\neg P$	...
...	...	...	...	...	

$\longrightarrow$

	0	1	2	3	...
$a_0 = a_2$	P	$\neg P$	P	$\neg P$	...
a_1	?	P	$\neg P$	?	...
...	...	...	...	...	

Figure 2 Trace of $\mathcal{M}^\infty$.

	0	1	2	3	...
d_0	P	$\neg P$	P	$\neg P$	...
d_1	?	P	$\neg P$	P	...
	$P(d_0) \wedge \mathbf{X}(\neg P(d_0))$	$P(d_1) \wedge \mathbf{X}(\neg P(d_1))$	$P(d_0) \wedge \mathbf{X}(\neg P(d_0))$	...	

Proof. Let α be a formula in $\mathrm{LTL}_{\Sigma,\{y\}}(\mathbf{X})$. We prove the theorem by induction over k_2.

If $k_1 = k_2$ then the result is reduced to lemma 21.

Induction step: we assume that the statement of the lemma holds for $[k_1, k_2]$. Now suppose that the premises of the lemma hold for $[k_1, k_2 + 1]$. From induction hypothesis, we know that there exists $\mathcal{M}_1$ satisfying the conclusion of the lemma for $i \in [k_1, k_2]$. We can then extend $\mathcal{M}^1$ by applying lemma 21 using instant $k_2 + 1$ and one element in A. Then the resulting structure satisfies the conclusion of the lemma for the set $[k_1, k_2 + 1]$. ◀

▶ **Example 23.** The main ideas of the proof of Lemma 22 are illustrated through an example. Let us consider the formula $\psi[y] = P(y) \wedge \mathbf{X}\neg P(y)$. For the sake of simplicity, instead of considering a finite temporal window $[k_1, k_2]$ among which $\exists y \cdot \psi[y]$ is satisfied, we consider a structure $\mathcal{M}$ s.t. for any $k \in \mathbb{N}$, $\mathcal{M}, k \models \exists y \cdot \psi[y]$, which is equivalent to $\mathcal{M}, 0 \models \mathbf{G}(\exists y \cdot \psi[y])$.

Let us build a finite partial model of this formula. Following the semantics of FOLTL, for any $k \in \mathbb{N}$, there is some a_k in the domain of $\mathcal{M}$ s.t. $\mathcal{M}, k, [y \mapsto a_k] \models P(y) \wedge \mathbf{X}\neg P(y)$. So, for any $k \in \mathbb{N}$, $P \in \rho_k(a_k)$ and $P \notin \rho_{k+1}(a_k)$. Consider the constraints a_0, a_1 and a_2 must satisfy: a_0 has constraints only at instants 0 (to satisfy P) and 1 (not to satisfy P); a_1 only has some constraints at instants 1 and 2; and a_2 only has some constraints at instants 2 and 3. Thus, we can reuse a_0 to play the role of a_2 at instants 2 and 3, as shown in Fig. 1.

Then ψ can be satisfied for the first three instants with only two elements in the domain. By the same argument, we can reuse a_1 instead of using a_3. This can be generalized to reuse a_0 (resp. a_1) instead of every a_k, where k is an even (resp. odd) number. We then see that we can satisfy our formula with a structure of size 2. Let us call d_0 and d_1 the corresponding elements of the domain. Let us define a first structure $\mathcal{M}^0 = (D, \sigma, \rho^0)$, where $D = \{d_0, d_1\}$, σ is an empty map (since there is no function symbols in ψ) and ρ^0 is defined as the partial function that is undefined over all entries. Now let us define $\mathcal{M}^{i+1}$ from $\mathcal{M}^i$. If i is even then $m = 0$, else $m = 1$ then Lemma 21 gives us $\mathcal{M}^{k+1} = \mathcal{M}^k[(k, d_m) \mapsto \{P\}][(k + 1, d_m) \mapsto \varnothing]$. Then we have $\mathcal{M}^{k+1}, k, [y \mapsto d_m] \models \psi[y]$.

We get a $\preccurlyeq$-increasing sequence $(\mathcal{M}^i)_{i\in\mathbb{N}}$, the limit structure of which $(\mathcal{M}^\infty)$ is illustrated in Fig. 2. Since for any integer k, $\mathcal{M}^{k+1}, k, [y \mapsto d_0] \models \psi[y]$ or $\mathcal{M}^{k+1}, k, [y \mapsto d_1] \models \psi[y]$, we have that $\mathcal{M}^\infty, k \models \mathbf{G}(\exists y \cdot \psi[y])$. ⌟

The reasoning that we had for this particular example can be easily generalized for any formula of the form $\mathbf{G}(\exists y \cdot \psi[y])$ where $\psi \in LTL_{\Sigma,\{y\}}(\mathbf{X})$. We then get a partial model of the formula with a domain of size $K_\psi + 1$.

Now, we want to extend the fragment to allow for the temporal connective $\mathbf{F}$ in ψ. Suppose that there is a model $\mathcal{M}$ of $\phi = \mathbf{G}(\exists y \cdot \psi[y])$ and that $\psi = \psi_1 \vee \ldots \vee \psi_n$ is in DNF, as in Lemma 18. Also suppose that several ψ_i have the form $\mathbf{F}\,\psi_i'$. Then, some of these $\mathbf{F}\,\psi_i'$ can be true at a finite number of instants in $\mathcal{M}$, which makes it complicated to build a finite partial model of ϕ. The following lemma states that we can get rid of such $\mathbf{F}\,\psi_i'$.

▶ **Lemma 24.** *Let $\mathcal{M}$ be a partial structure satisfying $\mathcal{M}, 0 \vDash \mathbf{G}(\psi_1 \vee \psi_2) \wedge \neg\,\mathbf{G}\,\mathbf{F}(\psi_2)$. Then there exists $\mathcal{M}'$ s.t. $\mathcal{M}', 0 \vDash \mathbf{G}(\psi_1)$ and $\mathcal{M}' \overset{Id}{\hookrightarrow} \mathcal{M}$, with Id defined as $Id(i,d) = d$ for all instant i and domain element d.*

Sketch. To get $\mathcal{M}'$ from $\mathcal{M}$, we simply make a translation in time, starting from the first instant k s.t. for any $k' \geqslant k$, $\mathcal{M}, k' \vDash \neg\psi_2$. ◀

4 Finite Domain Property

We now present our main results. We start in Sect. 4.1 with the BDP of our core fragment, limited to a single existential quantifier and without functions. In Sect. 4.2, we establish the BDP for extended fragments including functions and first-order quantifiers used in a restricted way. In Sect. 4.3, we study how these fragments can be extended with equality.

4.1 Core Theorem

Theorem 26 says that given a formula ϕ (1) in NNF, (2) with only one existential quantifier, (3) containing no other temporal connectives than $\mathbf{X}$ and $\mathbf{F}$, (4) without function symbols other than constants, (5) with only unary predicates, then $\mathbf{G}\,\phi$ enjoys the BDP. Most of these restrictions are unnecessary for the BDP but, while keeping the main ideas of the proof, they make it simpler to understand. Releasing them will lead to Theorem 28.

▶ **Definition 25.** *We say that $\phi \in Gur^{-}(\mathbf{X}, \mathbf{F})$ (for "Gurevich") if there exists a signature $\Sigma = (\mathcal{F}, \mathcal{R})$ s.t. (1) for any $n > 0$, $\mathcal{F}_n = \varnothing$, (2) for any $n > 1$, $\mathcal{R}_n = \varnothing$ and (3) there exists $\psi \in LTL_{\Sigma,\{y\}}(\mathbf{X}, \mathbf{F})$ s.t. $\phi = \exists y \cdot \psi$.*

▶ **Theorem 26.** *If ϕ is a formula in $Gur^{-}(\mathbf{X}, \mathbf{F})$, then $\mathbf{G}\,\phi$ enjoys the FDP. Moreover, if $\mathbf{G}\,\phi$ is satisfiable, it has a model of size $|Const| + 2 \times (K_\psi + 1)$.*

Proof. Let us consider that $\psi' \in \mathrm{LTL}_{\Sigma,\{y\}}(\mathbf{X}, \mathbf{F})$. By using Lemma 18, w.l.o.g., we consider ψ' in DNF: $\psi' = \psi_1 \vee \ldots \vee \psi_m$. Considering a model $\mathcal{M}$ of $\mathbf{G}(\exists y \cdot \psi'[y])$, some ψ_i are satisfied at an infinite number of instants. Let us consider that $\psi_1, \ldots, \psi_n$ are satisfied at an infinite number of instants and $\psi_{n+1}, \ldots, \psi_m$ are satisfied at a finite number of instants. Then by application of Lemma 24, there is a structure $\mathcal{M}$ satisfying: $\mathcal{M}, 0 \vDash \mathbf{G}(\exists y \cdot \psi_1[y] \vee \ldots \vee \psi_n[y])$.

Let us write $\psi = \psi_1 \vee \ldots \vee \psi_n$ and remind that each ψ_i is in NNF and each ψ_i is of the form $\alpha_i \wedge \mathbf{F}\,\beta_{i,1} \wedge \ldots \wedge \mathbf{F}\,\beta_{i,j_i}$ where $\alpha_i = \mathbf{X}^{n_{i,1}}\,\ell_{i,1} \wedge \ldots \wedge \mathbf{X}^{n_{i,k_i}}\,\ell_{i,k_i}$.

We introduce $\alpha = \bigvee_{\ell=1}^{n} \alpha_\ell$ and $\beta = \bigwedge_{\ell=1}^{n} \bigwedge_{p=1}^{j_\ell} \mathbf{F}\,\beta_{\ell,p}$.

The main step of the proof consists in defining a sequence $(\mathcal{M}^i, f^i, k_i)_{i \in \mathbb{N}}$ where, for each $i \in \mathbb{N}$:

- $\mathcal{M}^i$ is a partial structure, $\mathcal{M}^i \overset{f^i}{\hookrightarrow} \mathcal{M}$ and $\mathcal{M}^i \preccurlyeq \mathcal{M}^{i+1}$,
- up to instant $k_i - 1$, $\mathcal{M}^{i+1}$ coincides with $\mathcal{M}^i$,
- $\mathcal{M}^i$ is built as an extension of $\mathcal{M}^{i-1}$ s.t. for each $k < k_{i-1}$ $\mathcal{M}^i, k \models \exists y \cdot \psi[y]$,
- the limit $\mathcal{M}^\infty$ satisfies $\mathbf{G}(\exists y \cdot \psi[y])$ at instant 0.

The domain $\mathcal{D}$ of the different structures consists of the union of the two disjoints sets $\mathcal{D}_{\mathbf{X}} = \{d_0, \ldots, d_{K_\psi}\}$ and $\mathcal{D}_{\mathbf{F}} = \{e_0, \ldots e_{K_\psi}\}$, and of the set *Const* of constants. That is, $\mathcal{D} = \mathcal{D}_{\mathbf{X}} \cup \mathcal{D}_{\mathbf{F}} \cup Const$.

For $i = 0$, $\mathcal{M}^0$ and the partial function f^0 are defined by: (1) for any $k \in \mathbb{N}$ and $a \in \mathcal{D}_{\mathbf{X}} \cup \mathcal{D}_{\mathbf{F}}$, $f^0_k(a) = \bot$; and (2) $\mathcal{M}^0 \overset{f^0}{\hookrightarrow} \mathcal{M}$.

For any $i > 0$, let us now define $\mathcal{M}^i, k_i$ and f^i. $\mathcal{M}^i = (\mathcal{D}, \sigma^i, \rho^i)$ is defined as an extension of $\mathcal{M}^{i-1}$ in the following way. By application of Lemma 21, it is possible to extend $\mathcal{M}^{i-1}$ up to an instant k_i and satisfy β for one value of the domain. Within the time interval $[k_{i-1}, k_i[$, if i is an odd (resp. even) number, then $\mathcal{M}^i$ is s.t. β is satisfied at instant k_{i-1} for any $a \in \mathcal{D}_{\mathbf{F}}$ (resp. any $a \in \mathcal{D}_{\mathbf{X}}$): $\mathcal{M}^i, k_{i-1}, [y \mapsto a] \models \beta[y]$. If i is an odd (resp. even) number, this defines how $\mathcal{M}^i$ extends $\mathcal{M}^{i-1}$ for elements in $\mathcal{D}_{\mathbf{F}}$ (resp. $\mathcal{D}_{\mathbf{X}}$).

Now, by Lemma 22, if i is an odd (resp. even) number, we can extend $\mathcal{M}^{i-1}$ s.t. for any k in $[k_{i-1}, k_i[$, there is some $a \in \mathcal{D}_{\mathbf{X}}$ (resp. $a \in \mathcal{D}_{\mathbf{F}}$) s.t. $\mathcal{M}^i, k, [y \mapsto a] \models \alpha[y]$. If i is an odd (resp. even) number, this defines how $\mathcal{M}^i$ extends $\mathcal{M}^{i-1}$ for elements in $\mathcal{D}_{\mathbf{X}}$ (resp. $\mathcal{D}_{\mathbf{F}}$). Following this definition, for any $i \in \mathbb{N}$ and any $k < k_i$, $\mathcal{M}^i, k \models \exists y \cdot \psi[y]$.

The limit structure $\mathcal{M}^\infty$ of $(\mathcal{M}^i)_{i \in \mathbb{N}}$ is then a partial model of $\mathbf{G}(\exists y \cdot \psi[y])$, and its domain $\mathcal{D}$ is finite, of size $|Const| + 2 \times (K_\psi + 1)$. ◀

4.2 Relaxing the Use of Quantifiers

The next theorem generalizes the previous result to formulas:

- over n-ary predicates,
- with function symbols,
- and containing any number of existential quantifiers.

▶ **Definition 27.** *We say that $\phi \in Gur(\mathbf{X}, \mathbf{F})$ if there exists a signature Σ and a formula $\psi \in LTL_{\Sigma,\{y_1,\ldots y_n\}}(\mathbf{X}, \mathbf{F})$ such that $\phi = \exists y_1 \ldots y_n \cdot \psi$.*

▶ **Theorem 28.** *Given a formula ϕ in $Gur(\mathbf{X}, \mathbf{F})$, $\mathbf{G}\,\phi$ enjoys the BDP. Denoting $\mathcal{T}_\phi$ the set of terms appearing in ϕ, then, if $\mathbf{G}(\phi)$ is satisfiable, it has a model of size $|\mathcal{T}_\phi \cap \mathcal{T}_{\Sigma,\varnothing}| + 2 \times (K_\phi + 1) \times |\mathcal{T}_\phi \cap \mathcal{T}_{\Sigma,\mathcal{V}}|$.*

The fragment used in Theorem 28 forbids formulas outside the scope of $\mathbf{G}$. This prevents the specification of initial conditions. Proving the BDP for a fragment allowing such conditions requires to handle clauses in the DNF that are satisfied only a *finite* number of times, contrary to what we dealt with until then, using in particular Lemma 24. Theorem 31 states that we can actually extend the fragment used in Theorem 28 by adding a conjunct ψ to $\mathbf{G}(\phi)$ which refers to the initial state (and more generally to a finite set of states). However, the bound of the domain is significantly larger.

▶ **Definition 29.** *Let us assume that ϕ is in the form given in Lemma 18. Then we write $\beta_\phi = |\{\beta \mid \exists i \cdot \psi_i = \alpha_i \wedge \ldots \wedge \mathbf{F}\,\beta \wedge \ldots\}|$.*

▶ **Definition 30** (Genev fragment)**.** *We call* Genev fragment *the set of FOLTL formulas of shape $\psi \wedge \mathbf{G}(\phi)$ s.t. ϕ is a formula of class $Gur(\mathbf{X}, \mathbf{F})$ and $\psi = \exists y_1 \ldots y_2 \cdot \theta[y_1, \ldots, y_n]$ with $\theta \in LTL_{\Sigma,\{y_1,\ldots,y_n\}}$.*

▶ **Theorem 31.** *The Genev fragment enjoys the FDP. If $\psi \wedge \mathbf{G}(\phi)$ is a satisfiable formula in this fragment, it has a model of size $|(\mathcal{T}_\psi \cup \mathcal{T}_\phi) \cap \mathcal{T}_{\Sigma,\varnothing}| + (1 + 2^{\beta_\phi}) \times (K_\phi + 1) \times |\mathcal{T}_\phi \cap \mathcal{T}_{\Sigma,\mathcal{V}}|$.*

Let FO(∀) denote the fragment of purely universal FO formulas containing no other function symbols than constants. The next theorem extends Theorem 31 by allowing formulas of FO(∀) as leaves of the formula instead of basic predicates. However non-constant function symbols can not be used under the scope of a universal quantifier, since even the FO fragment of universally quantified formulas with non-restricted function symbols does not enjoy the FDP.

▶ **Definition 32.** *An FOLTL formula ψ is in FOLTL$(\exists\uparrow, \forall\downarrow)$ if $\psi = \exists y_1 \dots y_2 \cdot \theta[y_1, \dots, y_n]$, where θ has the following syntax: $\theta ::= \ell \mid \alpha \mid \theta \vee \theta \mid \theta \wedge \theta \mid \mathbf{X}\,\theta \mid \theta\,\mathbf{U}\,\theta \mid \theta\,\mathbf{R}\,\theta$, where $\alpha \in$ FO$(\forall)$ and ℓ is a literal.*

▶ Remark 33. Notice in particular that a formula in FOLTL$(\exists\uparrow, \forall\downarrow)$ satisfies the following two conditions: (1) no existential quantifier is in the scope of a temporal operator, (2) no temporal operator is in the scope of a universal quantifier. This is the case, for example, of the following formula: $\exists x, y \cdot (\forall z \cdot \neg P_1(z))\,\mathbf{U}\,(P_1(y)) \wedge (\forall z \cdot \neg P_2(x, z) \Rightarrow P_1(z))$.

▶ **Definition 34.** *FOLTL$(\mathbf{X}, \mathbf{F}, \forall\downarrow)$ is defined by the following grammar: $\phi ::= \ell \mid \alpha \mid \phi \vee \phi \mid \phi \wedge \phi \mid \mathbf{X}\,\phi \mid \mathbf{F}\,\phi \mid \exists y \cdot \phi$, with $\alpha \in$ FO$(\forall)$, ℓ a literal and $y \in \mathcal{V}$.*

▶ **Definition 35** (Geneva fragment)**.** *We call* Geneva fragment *the set of FOLTL formulas of shape $\psi \wedge \mathbf{G}(\phi)$ s.t. ϕ is a closed formula of FOLTL$(\mathbf{X}, \mathbf{F}, \forall\downarrow)$ and ψ is a closed formula of FOLTL$(\exists\uparrow, \forall\downarrow)$.*

▶ **Theorem 36.** *The Geneva fragment enjoys the FDP. If $\psi \wedge \mathbf{G}(\phi)$ is a satisfiable formula in this fragment, it has a model of size $|(\mathcal{T}_\psi \cup \mathcal{T}_\phi) \cap \mathcal{T}_{\Sigma,\varnothing}| + (1 + 2^{\beta_\phi}) \times (K_\phi + 1) \times |\mathcal{T}_\phi \cap \mathcal{T}_{\Sigma,\mathcal{V}}|$.*

4.3 Extension with Equality

We now address the problem of adding the equality predicate to the previous fragments. The interpretation of equality is constant over time. As mentioned in Sect. 3.1, this could be a source of infinity axioms if universal quantification is allowed. We show that we can add equality to the ∀-free fragments of our previous theorems 26, 28, and 31 and still enjoy the BDP. However, the bound on the domain becomes much larger and not exact anymore.

▶ **Definition 37.** *Given an FOLTL formula ϕ, we write $Eq(\phi)$ the set of equality tests of ϕ, i.e. the set of predicates of the form $t_1 = t_2$ in ϕ.*

In the following, Gur$^{=}(\mathbf{X}, \mathbf{F})$ (resp. LTL$^{=}_{\Sigma,\mathcal{V}}$) denotes Gur$(\mathbf{X}, \mathbf{F})$ (resp. LTL$_{\Sigma,\mathcal{V}}$) augmented with equality. Theorem 38 (resp. 39) generalizes Theorem 28 (resp. 31).

▶ **Theorem 38.** *If ϕ is a formula of class Gur$^{=}(\mathbf{X}, \mathbf{F})$ then $\mathbf{G}(\phi)$ enjoys the FDP. Writing $\mathcal{T}_\phi$ for the set of terms appearing in ϕ, then if $\mathbf{G}(\phi)$ is satisfiable, it has a model of size at most $|\mathcal{T}_\phi \cap \mathcal{T}_{\Sigma,\varnothing}| + 2 \times (K_\phi + 1) \times |\mathcal{T}_\phi \cap \mathcal{T}_{\Sigma,\mathcal{V}}| \times 2^{|Eq(\phi)|}$.*

Proof. Consider $\mathcal{M}$ a model of $\mathbf{G}(\phi)$ where $\phi = \exists \vec{y} \cdot \psi$. Theorem 28 can be applied to this formula after replacing equality tests by $\top$. This operation yields a partial structure that we call $\mathcal{M}_0$. Now we want to use $\mathcal{M}_0$ to build a model of $\mathbf{G}(\phi)$. Building such a model requires that, at each instant i, it is possible to find a tuple of elements in the domain that:

- satisfies the same relations as the tuple used to satisfy existential quantifiers at instant i in $\mathcal{M}_0$;
- satisfies the same equality relations as the tuple used to satisfy existential quantifiers at instant i in $\mathcal{M}$.

This can be done by making $2^{|Eq(\phi)|}$ copies of the domain of $\mathcal{M}_0$. Let us remember that this domain is an union of the tuples used to satisfy the existential quantifiers at different instants. Then, for each copy of each tuple, it is possible to define an equivalence relation between terms formed from this tuple. It requires to define these equivalence relations in order to cover all possibilities of interpretation for the equality relations appearing in ϕ, the number of possibilities being $2^{|Eq(\phi)|}$.

Once this is done, quotienting each part of the domain by this relation gives a structure where there are tuples:

- satisfying the same relations as any of the tuple of the first built finite model;
- satisfying any possible subset of $Eq(\phi)$.

So at any instant it is only needed to look in the original model what equality tests of the formula were satisfied and to take the tuple in the appropriate copy of the domain. ◀

▶ **Theorem 39.** *If ϕ is a formula of class $Gur^{=}(\mathbf{X},\mathbf{F})$ and $\psi = \exists y_1,\dots,y_n \cdot \theta[y_1,\dots,y_n]$, where $\theta \in LTL^{=}_{\Sigma,\{y_1,\dots,y_n\}}$, then $\psi \wedge \mathbf{G}(\phi)$ enjoys the FDP. If $\psi \wedge \mathbf{G}(\phi)$ is satisfiable, it has a model of size at most $|(\mathcal{T}_\psi \cup \mathcal{T}_\phi) \cap \mathcal{T}_{\Sigma,\varnothing}| + (1 + 2^{\beta_\phi}) \times 2^{|Eq(\phi)|} \times (K_\phi + 1) \times |\mathcal{T}_\phi \cap \mathcal{T}_{\Sigma,\mathcal{V}}|$.*

Proof. The proof of Theorem 38 can easily be adapted to Theorem 39. ◀

Now, if we extend the fragment of Theorem 36 with equality, it becomes possible to use equality predicates in the scope of a universal quantifier. In that case, our approach does not stand anymore. Therefore, the question of generalizing Theorem 36 by adding equality remains open.

5 Toy Example: a Notification System

Here, a simple example of a notification system in a ring is presented to illustrate the expressiveness of the fragment[4]. The structure of the network is defined by a predicate `succ` relating a node to its successor, and a formula **Ring** specifying that `succ` forms a ring topology. The formula **Ring** is specified as proposed in [15], with the use of a ternary predicate, in pure FO (without transitive closure). Each node x of the ring may be aware ($\texttt{notified}(x)$) of a certain piece of information, or not. Any node that has been notified may notify its successor (by *sending* a message), and other nodes do not change during this operation.

$$\begin{aligned}
\mathbf{Same}(z) &:= \texttt{notified}(z) \Leftrightarrow \mathbf{X}\,\texttt{notified}(z)\\
\mathbf{Send}(x) &:= \exists y \cdot \Big[\texttt{succ}(x,y) \wedge (\forall z \cdot z \neq y \Rightarrow \mathbf{Same}(z))\\
&\qquad \wedge (\texttt{notified}(x) \Rightarrow \mathbf{X}\,\texttt{notified}(y)) \wedge (\neg\texttt{notified}(x) \Rightarrow \mathbf{Same}(y))\Big]\\
\mathbf{Trans} &:= \mathbf{G}(\exists p \cdot \mathbf{Send}(p))
\end{aligned}$$

5.1 Safety Property

Now consider the safety property "if a node is notified, it remains notified", described by the following formula: $\mathbf{Safety} := \mathbf{G}(\forall x \cdot \texttt{notified}(x) \Rightarrow \mathbf{X}\,\texttt{notified}(x))$. Proving that our protocol ensures this property ($\mathbf{Ring} \wedge \mathbf{Trans} \models \mathbf{Safety}$) amounts to proving that $\mathbf{Ring} \wedge \mathbf{Trans} \wedge \neg\mathbf{Safety}$ is unsatisfiable.

[4] The complete Electrum specification is available at [16]. Electrum is available at `http://huit.re/electrum/`.

Notice that $\neg\mathbf{Safety} \equiv \exists x \cdot \mathbf{F}(\mathtt{notified}(x) \wedge \mathbf{X}\,\neg\mathtt{notified}(x))$, therefore an equisatisfiable formula can be obtained by Skolemization: $\mathbf{SkNegSafety} := \mathbf{F}(\mathtt{notified}(c) \wedge \mathbf{X}\,\neg\mathtt{notified}(c))$. However $\varphi := \mathbf{Ring} \wedge \mathbf{Trans} \wedge \mathbf{SkNegSafety}$ is not in any of our fragments because of the universal quantification over $\mathbf{Same}(z)$, which is a temporal formula, and because of the use of equality.

We now devise a more abstract specification of the protocol which is a semantic consequence of φ that fits into the fragment of Theorem 36. First, we get rid of equality: we use an equivalence predicate $\approx$ instead, which can be axiomatized (using a formula $\overline{\mathbf{Eq}}$) in our fragment (notice that the semantics of $\approx$ may vary over time). Second, we get rid of the universal quantifier over z. To do that, a solution is to instantiate the variable z for the values x and c, which yields: $\overline{\mathbf{Send}}(x) := (\exists y \cdot \mathtt{succ}(x,y) \wedge (\mathtt{notified}(x) \Rightarrow \mathbf{X}\,\mathtt{notified}(y)) \wedge (\neg\mathtt{notified}(x) \Rightarrow \mathbf{Same}(y)) \wedge (x \approx y \vee \mathbf{Same}(x)) \wedge c \approx y \vee \mathbf{Same}(c)) \wedge (c \approx y \Leftrightarrow \mathbf{X}\,c \approx y))$. Notice it is necessary to add $c \approx y \Leftrightarrow \mathbf{X}\,c \approx y$ in the previous formula. Indeed, $\approx$ is not necessarily constant so it would be possible to have that $c \approx y$ and $\neg\,\mathbf{X}\,c \approx y$ and, in this case, no constraint would apply to the truth value of $\mathbf{X}\,\mathtt{notified}(c)$. Then, it is possible to define an abstraction by the following formulas: $\overline{\mathbf{Trans}} := \mathbf{G}(\exists p \cdot \overline{\mathbf{Send}}(p))$ and $\mathbf{AbsSatS} := \overline{\mathbf{Eq}} \wedge \overline{\mathbf{Trans}} \wedge \mathbf{SkNegSafety}$.

It is easy to show that $\mathbf{Ring} \wedge \mathbf{Trans} \wedge \neg\mathbf{Safety} \models \mathbf{AbsSatS}$ and that $\mathbf{AbsSatS}$ belongs to the Geneva fragment. Applying Theorem 36, we compute a size 5 for the domain. Using the Electrum tool, $\mathbf{AbsSatS}$ can be shown to be unsatisfiable for a domain of size 5, which ultimately proves the original property.

5.2 Liveness Property

An interesting liveness property to prove on the considered system is "all nodes eventually become notified", formalized as: $\mathbf{Liveness} := \forall x \cdot \mathbf{F}(\mathtt{notified}(x))$. This property can be shown under the assumption that all notified nodes eventually perform the send transition: $\mathbf{Progress} := \mathbf{G}(\forall x \cdot \mathtt{notified}(x) \Rightarrow \mathbf{F}\,\mathbf{Send}(x))$.

The complete abstraction that allows us to prove this liveness property is available with the full example specification. We basically need to Skolemize the negation of the liveness property and to instantiate the universally quantifiers that are out of our fragment with the Skolem constant. An axiom abstracting the ring topology needs to be added for proving the liveness property. In the end, the obtained formula fits into the fragment of Theorem 36, which provides the domain size 6. The formula can be shown in Electrum to be unsatisfiable for a domain of size 6, which proves the property.

6 Related Work

In [8], Kuperberg and the last two authors of the present article show that the FDP for some FO fragments can be lifted to some FOLTL fragments. However, they only allow to add $\mathbf{X}$ and $\mathbf{F}$ connectives, which is not enough for real specifications. An extension of the Ramsey fragment is also proposed, allowing the use of all temporal connectives, but preventing existential quantifiers under a $\mathbf{G}$.

The decidable monodic fragment studied by Hodkinson et al. [5,6] does not enjoy the FDP. Indeed, $\mathbf{G}(\exists y \cdot P(y) \wedge \mathbf{G}(\neg P(y)))$ belongs to the monodic extension of the Gurevich fragment (first-order formulas containing existential quantifiers only) but it is an axiom of infinity: the monodic fragment helps preserve decidability but says nothing about the FDP. Additionally, on the practical side, the monodic fragment limits the use of free variables in temporal formulas to only one, which does not really fit with real specifications of systems. Indeed,

any transition system implying relations between different components (list of messages, topology of a network, etc) requires to be specified by using at least binary relations in the temporal transitions, thus breaking the monodicity condition.

Padon et al. [15] propose yet another approach: they reduce specific temporal problems to FO and even, in many cases, to a *decidable* fragment of it. This method was improved in [13, 14] to address the verification of liveness properties. It was implemented in the Ivy tool and gives good results in practice. However, it is not complete and it requires the user to understand rather deeply both the specified system and the verification technique itself. Additionally, the user must devise an inductive invariant manually.

7 Discussion

In the introduction, we drew as an inspiration for our work the following classical shape for specifications of systems and of their properties: $\mathit{spec} = \mathit{init} \wedge \mathbf{G}\,\mathit{trans} \wedge \mathit{fair} \rightarrow \mathit{prop}$ (with *trans* using only the $\mathbf{X}$ connective). Checking the validity of *spec* amounts to assessing the satisfiability of $\neg\mathit{spec} = \mathit{init} \wedge \mathbf{G}\,\mathit{trans} \wedge \mathit{fair} \wedge \neg\mathit{prop}$. Our results then say this satisfiability can be decided provided $\neg\mathit{spec}$ can written as $\psi \wedge \mathbf{G}(\phi)$ and respect the conditions of Theorems 31, 36 or 39.

Beyond the obvious *init* and *trans*, one can see that, depending on their shape, *fair* and *prop* will have to be, possibly, split into sub-formulas, and then "dispatched" into either ψ or ϕ, or both. As an example, for *prop*, any combination of an $\mathbf{F}$ or $\mathbf{G}$ connective and of some quantification on a variable is acceptable, except for the shape $\exists x \cdot \mathbf{G}(P(x))$ (as we would have $\neg\mathit{prop} = \forall x \cdot \mathbf{F}(\neg P(x))$, in which case $\forall$ would not appear as a leaf). Similarly, for a strong fairness property of the shape $\mathbf{G}\,\mathbf{F}\,\mathit{enabled} \rightarrow \mathbf{G}\,\mathbf{F}\,\mathit{effect} = \mathbf{F}\,\mathbf{G}\,\neg\mathit{enabled} \vee \mathbf{G}\,\mathbf{F}\,\mathit{effect}$, the disjunction distributes over the rest of the formula, and then *enabled* may only contain existential quantifiers at the leaves and universal ones at the root, and *effect* may only have universal quantifiers at the leaves (without any constraint on existential ones). This may sometimes be restrictive, in which case alternative expressions should be sought. Another limitation lies in the possible uses of (constant) equality: in our first experiments, we were often able to abstract it into a dynamic equivalence relation, as we did in Sect. 5.

Now, if the specification falls into one of our fragments, then the bound on the domain is known (and even exact, without equality). To be sure, this bound grows exponentially but only in the number of $\mathbf{F}$ connectives under a $\mathbf{G}$. This ultimately yields a decision procedure for the validity of *spec*. Notice that existing tools, such as our own Electrum [10], can readily be used to support it, as was shown in Sect. 5.

In the future, we will study ways to augment the expressiveness of our fragments to address some of the current limitations (e.g. fairness, equality). Apart from trying to extend the fragments themselves, we will also devise a many-sorted version thereof, in the spirit of [11, 12, 1, 15], to extend their expressiveness and applicability and to fit the data structuring features of Electrum [10] more closely. We will also assess our approach on more realistic case studies. Finally, we will build on our results in the setting of complete, automated verification for system specification.

References

1 Aharon Abadi, Alexander Rabinovich, and Mooly Sagiv. Decidable fragments of many-sorted logic. *Journal of Symbolic Computation*, 45(2):153–172, 2010. doi:10.1016/j.jsc.2009.03.003.

2 Julien Brunel, David Chemouil, Alcino Cunha, and Nuno Macedo. The Electrum Analyzer: Model Checking Relational First-Order Temporal Specifications. In *33rd ACM/IEEE International Conference on Automated Software Engineering (ASE '18)*, Montpellier, France, September 2018. ACM Press. doi:10.1145/3238147.3240475.

3 Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997. doi:10.1007/978-3-642-59207-2.

4 Dov M. Gabbay, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*, chapter Fragments of first-order temporal logics. Elsevier, 2003.

5 Ian Hodkinson, Frank Wolter, and Michael Zakharyaschev. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied logic*, 106(1-3):85–134, 2000. doi:10.1016/S0168-0072(00)00018-X.

6 Ian Hodkinson, Frank Wolter, and Michael Zakharyaschev. Monodic Fragments of First-Order Temporal Logics: 2000–2001 A.D. In *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 1–23. Springer, 2001. doi:10.1007/3-540-45653-8_1.

7 Fred Kröger and Stephan Merz. *Temporal Logic and State Systems (Texts in Theoretical Computer Science. An EATCS Series)*. Springer, 2008.

8 Denis Kuperberg, Julien Brunel, and David Chemouil. On Finite Domains in First-Order Linear Temporal Logic. In *Automated Technology for Verification and Analysis*, pages 211–226. Springer, 2016. doi:10.1007/978-3-319-46520-3_14.

9 Leslie Lamport. *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley Professional, 2002.

10 Nuno Macedo, Julien Brunel, David Chemouil, Alcino Cunha, and Denis Kuperberg. Lightweight Specification and Analysis of Dynamic Systems with Rich Configurations. In *Foundations of Software Engineering*, Seattle, United States, November 2016. doi:10.1145/2950290.2950318.

11 Timothy Nelson, Daniel J. Dougherty, Kathi Fisler, and Shriram Krishnamurthi. On the finite model property in order-sorted logic. Technical report, Worcester Polytechnic Institute, 2010.

12 Timothy Nelson, Daniel J. Dougherty, Kathi Fisler, and Shriram Krishnamurthi. Toward a More Complete Alloy. In *Abstract State Machines, Alloy, B, VDM, and Z*, pages 136–149. Springer, 2012. doi:10.1007/978-3-642-30885-7_10.

13 Oded Padon, Jochen Hoenicke, Giuliano Losa, Andreas Podelski, Mooly Sagiv, and Sharon Shoham. Reducing liveness to safety in first-order logic. *Proceedings of the ACM Conference on Principles of Programming Languages (POPL)*, 2:26, 2017. doi:10.1145/3158114.

14 Oded Padon, Jochen Hoenicke, Kenneth L McMillan, Andreas Podelski, Mooly Sagiv, and Sharon Shoham. Temporal Prophecy for Proving Temporal Properties of Infinite-State Systems. In *Formal Methods in Computer Aided Design (FMCAD)*, 2018. doi:10.23919/FMCAD.2018.8603008.

15 Oded Padon, Kenneth L. McMillan, Aurojit Panda, Mooly Sagiv, and Sharon Shoham. Ivy: safety verification by interactive generalization. *ACM SIGPLAN Notices*, 51(6):614–630, 2016. doi:10.1145/2980983.2908118.

16 Quentin Peyras, Julien Brunel, and David Chemouil. Electrum specification of a toy notification system, August 2019. doi:10.5281/zenodo.3369542.