A Decidable and Expressive Fragment of Many-Sorted First-Order Linear Temporal Logic

Quentin Peyras, Julien Brunel, David Chemouil

A Decidable and Expressive Fragment of Many-Sorted First-Order Linear Temporal Logic

Quentin Peyras, Julien Brunel, David Chemouil*

DTIS, ONERA & Université fédérale de Toulouse, F-31055 Toulouse

Abstract

First-Order Linear Temporal Logic (FOLTL) and its Many-Sorted variant (MSFOLTL) are well-suited to specify infinite-state systems. However, the satisfiability of (MS)FOLTL is not even semi-decidable, thus preventing automated verification. In this paper, we exhibit various fragments of increasing scope that provide a pertinent basis for the abstract specification of infinite-state systems. We show that these fragments enjoy the Bounded Domain Property (any satisfiable (MS)FOLTL formula has a model with a finite, *bounded* FO domains), 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.

Keywords: First-Order Linear Temporal Logic, Many-Sorted Logic, Bounded Domain Property, Finite Domain Property, Decidability.
2010 MSC: 03B44, 03B70

1. Introduction

First-Order Logic (FO), even more so in its Many-Sorted variant, 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. (Many-Sorted) 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) [1, 2] is the most studied among those.

However, formally *verifying* properties of such specifications is challenging as (MS)FOLTL is not even semi-decidable. To overcome this situation, a solution is to restrict the expressive power of the temporal dimension of the logic by focusing on the verification of safety properties. It is then possible to rely on decidable (MS)FO fragments [3] to develop a verification procedure. This approach is for instance followed in Ivy [4]. When it comes to verifying liveness properties, an extension of Ivy relies on a liveness-to-safety transformation [5, 6]. These techniques are promising but they lead to an incomplete verification procedure and are not entirely automated.

Another approach, previously followed by the present authors, limits analyses to a bounded FO domain. The verification procedure is then entirely automated and has shown to be easily applicable for the quick validation of rich system specifications [7–9]. However, since only a bounded FO domain is explored, the verification procedure is also incomplete.

In this paper, we aim at verifying abstract specifications of infinite-state systems, expressed in (MS)FOLTL, automatically. So we follow yet another track: we exhibit *fragments* of (MS)FOLTL which we claim *expressive* enough to specify a large class of systems and which enjoy an *automated* and *complete* verification method. In practice, we prove that the fragments enjoy the *Bounded Domain Property* (BDP):

*Corresponding author.
Email addresses: quentin.peyras@onera.fr (Quentin Peyras), julien.brunel@onera.fr (Julien Brunel), david.chemouil@onera.fr (David Chemouil)

1. finite domain property (FDP): any satisfiable formula in these fragments has a model with finite FO domains,
2. we effectively exhibit a bound on the FO domains (depending on the size of the formula and often exact).

We stress that the bound does *not* apply to the *temporal* dimension of models.

Remark that this work builds upon previous work of some of the present authors [10], where various simple fragments of FOLTL were studied. None of those fragments allows to express a typical system specification. In particular, only one fragment including all LTL connectives, an extension of the classic Ramsey FO fragment (cf. Ex. 1), is shown to enjoy the FDP. This fragment is strictly included in the most general fragment presented in this paper, for which we establish the BDP.

Owing to our purpose, the syntactic shape of our fragments is inspired by formal system specification approaches such as Lamport's TLA$^+$ [11] or the present authors' Electrum [7, 8] (both these languages having a strong relation with MSFOLTL). In Electrum for instance, an abstract system specification *spec* typically has the following shape (ignoring additional features of the language):

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

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

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

Our main contribution is precisely to devise some novel *decidable* fragments of (MS)FOLTL encompassing formulas following the $\neg spec$ template, by showing that these fragments enjoy the BDP.

Thus, provided $\neg spec$ belongs to one of these fragments, there is for every sort S a bound k_S such that the problem $\mathrm{UNSAT}_{\mathrm{MSFOLTL}}(\neg spec)$ can be reduced to $\mathrm{UNSAT}^{\bigwedge_{S \in \mathcal{S}} |S| \leqslant k_S}_{\mathrm{MSFOLTL}}(\neg spec)$, where this notation means unsatisfiability in interpretation structures where, for every sort S, the corresponding FO domain is of size at most k_S.

Finally, using these bounds, the MSFOLTL formula *spec* can be expanded into a plain LTL formula *spec*$'$, by unfolding quantifiers over the bounded domains. This way, the $\mathrm{UNSAT}^{\bigwedge_{S \in \mathcal{S}} |S| \leqslant k_S}_{\mathrm{MSFOLTL}}(\neg spec)$ problem is itself reduced to the problem $\mathrm{UNSAT}_{\mathrm{LTL}}(\neg spec')$. 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 bound is *effectively reached*, in the sense that $\mathrm{UNSAT}_{\mathrm{MSFOLTL}}(\neg spec)$ can even be reduced to $\mathrm{UNSAT}^{\bigwedge_{S \in \mathcal{S}} |S| = k_S}_{\mathrm{MSFOLTL}}(\neg spec)$, which can in practice be leveraged to produce a smaller LTL formula to check for unsatisfiability.

Comparison with [12]. This article extends work presented at TIME'19. Compared with the former work, the present article details many proofs in Sect. 4 and introduces a whole new *many-sorted* extension of former results in Sect. 5. In this section, we present a novel fragment of MSFOLTL enjoying the BDP. Remark that this fragment generalizes both the Geneva fragment of Sect. 4 and the last fragment presented in [10]. Additionally, in this version of the article, the example presented in Sect. 6 has been updated to take into account the many-sorted setting.

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

a notion of *partial* interpretation structures that is helpful in building interpretation structures step-by-step rather than in one stroke. In Sect. 4, we define in several steps a fragment of FOLTL that is relevant in the context of system specification. We show that it enjoys the BDP and exhibit a bound on the FO domain. In Sect. 5 we define an extension of the previous fragment in many-sorted logic and prove that this fragment enjoys the BDP. We also provide an algorithm computing the corresponding bound. Then, in Sect 6, we illustrate the interest and limitations of our many-sorted fragment on a toy example. Finally, we draw a comparison with related work in Sect. 7.

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 $\mathcal{T}_{\Sigma,\mathcal{V}}$):*

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

X and **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 **X** (resp. **X** and **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 1 (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. 6) and proofs.*

Definition 3 (Assignment)**.** *An* assignment *C in a domain D for variables in $\mathcal{V}$ is a map $\mathcal{V} \to D$. We write $C[x \mapsto d]$ the assignment defined as $C[x \mapsto d](x) = d$ and $C[x \mapsto d](y) = C(y)$ if $y \neq x$. The* extension *of C to* terms*, also written C, is defined in the obvious way.*

Definition 4 (Satisfaction)**.** *Given a structure $\mathcal{M} = (D, \sigma, \rho)$ and an assignment C, the* satisfaction *relation $\vDash$ is defined by induction on formulas, for any $i \in \mathbb{N}$, as follows:*

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

Given a closed formula ϕ, we write $\mathcal{M}, k \vDash \phi$ if $\mathcal{M}, k, [] \vDash \phi$, where $[]$ is the empty assignment. A structure $\mathcal{M}$ is a model *of ϕ if $\mathcal{M}, 0 \vDash \phi$. A formula that has at least one model is said to be* satisfiable*.*

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

Definition 5 (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 2 (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 3 (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 [10]. Indeed, in [10], the complexity of the satisfiability problem on bounded models is studied for full FOLTL.*

Example 1. *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. Axioms of Infinity

There are various ways not to enjoy the FDP. 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.

Inspired by results from [10], 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.

Thus, existential quantification under a $\mathbf{G}$ connective can be problematic. However, this only happens when several *nested* $\mathbf{G}$ connectives appear in a formula, which is rarely necessary in practical system specifications.

Therefore, we 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.

First, 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 situation is actually similar to the first axiom. In order to avoid this behaviour, an additional restriction is needed: one possibility is to forbid the use of temporal connectives in 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 a 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 already-used elements. Satisfying the formula then requires to have an infinite domain.

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

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

3. Partial Structures

In the previous section, 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 interpretations of predicates 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 would need to redefine the entire structure at each step. For this reason, in Sect. 3.1, we introduce a notion of partial structures, which is easier to handle. Then, in Sect. 3.2, we provide some technical lemmas relying on partial structures, which will be useful to establish the BDP of the fragments studied in Sect. 4.

3.1. Definitions

Definition 6 (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 7 (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: a partial structure satisfies a formula if *all* its extensions that are structures satisfy it.

Definition 8 (Semantics over partial structures I)**.** *Given a partial structure $\mathcal{M}$, we say that $\mathcal{M}, i, C \models_p \phi$ iff for all structure $\mathcal{M}'$ s.t. $\mathcal{M} \preccurlyeq \mathcal{M}'$, we have $\mathcal{M}', i, C \models \phi$. Similarly to Def 4, Given a closed formula ϕ, we write $\mathcal{M}, k \models_p \phi$ if $\mathcal{M}, k, [] \models_p \phi$, where [] is the empty assignment. We also say that $\mathcal{M}$ is a partial model of ϕ if $\mathcal{M}, 0 \models_p \phi$.*

There is another, natural way to define the semantics over partial structures, which 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. Indeed, 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 9 (Semantics over partial structures II)**.** *Given a partial structure $\mathcal{M} = (D, \sigma, \rho)$ and an assignment C, the* satisfaction *relation $\Vdash$ is defined by induction on formulas in negation normal form (NNF), for all non-negative integers i as follows:*

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

Lemma 1 (Equivalence of semantics)**.** *Given a partial structure $\mathcal{M}$, a formula ϕ in NNF, $k \in \mathbb{N}$ and an assignment C, we have $\mathcal{M}, k, C \models_p \phi$ iff $\mathcal{M}, k, C \Vdash \phi$.*

Proof. By induction over formulas:

Consider a literal $l = r(t_1, \ldots t_n)$, a partial structure $\mathcal{M} = (D, \sigma, \rho)$, an assignment C and an integer i, such that $\mathcal{M}, i, C \Vdash r(t_1, \ldots, t_n)$. Given an arbitrary structure $\mathcal{M}' = (D, \sigma, \rho')$ extending $\mathcal{M}$, we have $r \in \rho'_i(C(t_1), \ldots, C(t_n))$ since $\rho_i(C(t_1), \ldots, C(t_n))$ is defined and since $r \in \rho_i(C(t_1), \ldots, C(t_n))$. Therefore $\mathcal{M}', i, C \models_p r(t_1, \ldots, t_n)$ and then, by definition of $\models_p$ for partial structures, $\mathcal{M}, i, C \models_p r(t_1, \ldots, t_n)$.

In the other direction, assume that $\mathcal{M}, i, C \models_p r(t_1, \ldots, t_n)$. Suppose that $\mathcal{M}, i, C \nVdash r(t_1, \ldots, t_n)$. Then either $\rho_i(C(t_1), \ldots, C(t_n))$ is undefined, or $r \notin \rho_i(C(t_1), \ldots, C(t_n))$: in both cases, we can consider a structure $\mathcal{M}' = (D, \sigma, \rho')$ such that r is not in $\rho'_i(C(t_1), \ldots, C(t_n))$. Therefore $\mathcal{M}, i, C \not\models r(t_1, \ldots, t_n)$. Contradiction. Therefore $\mathcal{M}, i, C \Vdash r(t_1, \ldots, t_n)$.

Consider now the case $l = \neg r(t_1, \ldots t_n)$. The proof is similar to the previous case. Assume that $\mathcal{M}, i, C \Vdash r(t_1, \ldots, t_n)$. Considering an arbitrary extension of $\mathcal{M}$, we conclude that $\mathcal{M}, i, C \models_p \neg r(t_1, \ldots, t_n)$. Suppose now that $\mathcal{M}, i, C \not\models r(t_1, \ldots, t_n)$. Then, by the same argument, the fact that $\mathcal{M}, i, C \not\models \neg r(t_1, \ldots, t_n)$ implies that there exists a structure $\mathcal{M}' = (D, \sigma, \rho')$ extending $\mathcal{M}$ and s.t. $r \in \rho'_i(C(t_1), \ldots, C(t_n))$. Which is impossible. Therefore $\mathcal{M}, i, C \models_p r(t_1, \ldots, t_n)$.

The rest of the proof trivially follows from a simple application of the definition of the semantics for each connective or quantifier. □

As we focus on the BDP, we aim at building a domain-finite model of a formula out of any structure satisfying it (an illustration is developed in Example 2). To do so, we will follow an iterative procedure, enriching a partial structure. This enrichment is defined as follows.

Definition 10 (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 $\mathcal{M}$.*

After applying the iterative procedure which enriches a partial structure, we get an increasing sequence of partial structures. Intuitively, such a sequence somehow converges to a partial structure on which all steps of extension have been performed. The following definition formalizes this notion.

Definition 11 (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. $\rho^k_i(\vec{d}) \neq \bot$, then $\rho^\infty_i(\vec{d}) = \rho^k_i(\vec{d})$; (2) if for every $k \in \mathbb{N}$ we have $\rho^k_i(\vec{d}) = \bot$, then $\rho^\infty_i(\vec{d}) = \bot$.*

However, to ensure that we have a general method working for a fragment that is 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 *shifting embedding*. Informally, this embedding between two partial structures expresses that any element of the domain of the former has, for each instant, a corresponding 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 corresponding elements are considered corresponding tuples, so they satisfy the same predicates. Also remark that this embedding allows for a constant time shift between the two structures, represented by the integer m. Similar to the case of partial structures, the shifting embedding must be *partial* both over the domain and over time. Indeed, the proofs require to build structures step by step, by defining values of predicates over part of the time and part of the domain. Since the shifting embedding must be defined alongside a partial structure, it is also necessary to define it step by step.

Definition 12 (Shifting 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* (shifting) 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. $f_i(\vec{d}) \neq \bot$[1], $f_i(\sigma_0(g)(\vec{d})) = \sigma_1(g)(f_i(\vec{d}))$; and*
- *for each $\vec{d} \in D_0^\star$ and each $i \in \mathbb{N}$, if $f_{i+m}(\vec{d}) \neq \bot$ then $\rho^0_i(\vec{d}) = \rho^1_i(f_{i+m}(\vec{d}))$, otherwise $\rho^0_i(\vec{d}) = \bot$.*

Remark 4. *Remark that, provided $\mathcal{M}_1$, m and f are given, then the value of ρ_0 directly follows from the third bullet of the previous definition. This property is useful as it allows us to define a partial structure $\mathcal{M}_0$ only by giving the integer m and the shifting embedding f into a given structure $\mathcal{M}_1$. In practice, it allows to define a finite partial model of a formula by defining a shifting embedding from a finite domain into a given arbitrary model of this formula.*

3.2. Preliminary Lemmas

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.

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 rather study the *disjunctive normal form* of ϕ, 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 **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 **X** connectives).

[1] $f_i(d_1, \ldots, d_n)$ is defined as $(f_i(d_1), \ldots, f_i(d_n))$ if f_i is defined over $\{d_i \mid 1 \leq i \leq n\}$, otherwise $f_i(d_1, \ldots, d_n) = \bot$. Thus $f_i(\vec{d}) \neq \bot$ denotes the fact that $f_i(\vec{d})$ is defined.

Lemma 2 (Disjunctive normal form (DNF))**.** *Given a formula ϕ in $LTL_{\Sigma,\mathcal{V}}(\mathbf{X},\mathbf{F})$ 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 5 (Inocuity of the DNF)**.** *In this paper, the DNF is only used to prove the BDP for the considered fragments. On the other hand:*

- *the computed bounds are not affected by the DNF;*
- *and a given formula does not have to be in DNF to check whether it belongs to any of our fragments.*

Therefore, the exponential blow-up associated to DNF transformation does not *affect the decision procedure discussed in Sect. 1.*

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 13 (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 2).*

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 3. *Consider a formula ψ in $LTL_{\Sigma,\{y\}}(\mathbf{X},\mathbf{F})$ and a structure $\mathcal{M} = (\mathcal{D}_\mathcal{M}, \sigma, \rho)$ s.t. $\mathcal{M}, k \vDash_p \exists y \cdot \psi[y]$ 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 $i \in \mathbb{N}$ and any $x \in D$, $f^0_i(x) \neq f^1_i(x)$ iff $x = a$ and $k \leq i < k'$*
- *$\mathcal{M}_0 \preccurlyeq \mathcal{M}_1$,*
- *$\mathcal{M}_1, k, [y \mapsto a] \vDash_p \psi[y]$.*

Proof. Consider a structure $\mathcal{M}$ and a partial structure $\mathcal{M}_0$ satisfying the conditions of Lemma 3.

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 ([10]), in the sense that changing the interpretation of predicates outside I does not change the truth value of the formula. This can be shown by induction on the number of nested $\mathbf{F}$ connectives.

Now, let ψ be a formula in $\mathrm{LTL}_{\Sigma,\{y\}}(\mathbf{X},\mathbf{F})$ s.t. $\mathcal{M}, k \vDash \exists y \cdot \psi[y]$. Let k' be the greatest instant in the set I as introduced above. Let d be an element in the domain such that $\mathcal{M}, k, [y \mapsto d] \vDash \psi[y]$. Then, we can extend $\mathcal{M}_0$ into $\mathcal{M}_1$ in such a way that $f^1_i(a) = d$ (implying $\rho^1_i(a) = \rho_i(d)$) for $i \in [k, k'-1]$ (outside of this set ρ^1 and f^1 coincide respectively with ρ^0 and f^0). □

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 1. In particular, we limit the result to a finite temporal window $[k_1, k_2]$.

	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 1: First step of the partial structure construction.

Lemma 4. *Assume that there exists $k_1, k_2 \in \mathbb{N}$ s.t. for any integer $i \in [k_1, k_2]$ we have $\mathcal{M}, i \vDash_p \exists y \cdot \alpha[y]$, 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. Suppose 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.* $\mathcal{M}^1, i, [y \mapsto a_m] \vDash_p \alpha[y]$.

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 3.

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 the induction hypothesis, 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 3 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 2. *The main ideas of the proof of Lemma 4 are illustrated through an example. 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 $k \in \mathbb{N}$, $\mathcal{M}, k \vDash \exists y \cdot \psi[y]$, which is equivalent to $\mathcal{M}, 0 \vDash \mathbf{G}(\exists y \cdot \psi[y])$.*

We now 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. $\mathcal{M}, k, [y \mapsto a_k] \vDash P(y) \wedge \mathbf{X}\neg P(y)$. 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.

Thus, 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. 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 3 gives us $\mathcal{M}^{k+1} = \mathcal{M}^k[(k, d_m) \mapsto \{P\}][(k+1, d_m) \mapsto \varnothing]$. Then we have $\mathcal{M}^{k+1}, k, [y \mapsto d_m] \vDash_p \psi[y]$.

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, $\mathcal{M}^{k+1}, k, [y \mapsto d_0] \vDash_p \psi[y]$ or $\mathcal{M}^{k+1}, k, [y \mapsto d_1] \vDash_p \psi[y]$, we have $\mathcal{M}^\infty, k \vDash_p \mathbf{G}(\exists y \cdot \psi[y])$. □

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 2. Also suppose that several ψ_i have the

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

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

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

Proof 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 $k' \geqslant k$, $\mathcal{M}, k' \vDash_p \neg\psi_2$. □

4. Bounded 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 larger fragments including functions and first-order quantifiers used in a restricted way. In Sect. 4.4, we study how these fragments can be extended with equality.

4.1. Core Theorem

Theorem 1 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 2.

Definition 14. *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 1. *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. Consider a formula $\psi' \in \mathrm{LTL}_{\Sigma,\{y\}}(\mathbf{X}, \mathbf{F})$. By Lemma 2, we suppose w.l.o.g. that ψ' is in DNF, *i.e.* $\psi' = \psi_1 \vee \ldots \vee \psi_m$. Given a model $\mathcal{M}$ of $\mathbf{G}(\exists y \cdot \psi'[y])$, some ψ_i are satisfied at an infinite number of instants. Suppose 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, for some n. Then by Lemma 5, there is a structure $\mathcal{M}$ s.t. $\mathcal{M}, 0 \vDash_p \mathbf{G}(\exists y \cdot \psi_1[y] \vee \ldots \vee \psi_n[y])$.

We write $\psi = \psi_1 \vee \ldots \vee \psi_n$. Notice 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 define $\alpha = \bigvee_{\ell=1}^{n} \alpha_\ell$ and $\beta = \bigwedge_{\ell=1}^{n} \bigwedge_{p=1}^{j_\ell} \mathbf{F}\,\beta_{\ell,p}$. Remark that α and β are defined so that the satisfaction of $\alpha \wedge \beta$ implies the satisfaction of ψ. (This is the reason why β is not defined as $\bigvee_{\ell=1}^{n} \bigwedge_{p=1}^{j_\ell} \mathbf{F}\,\beta_{\ell,p}$. If it was the case, it would be possible to satisfy α_i (thus satisfying α) and $\bigwedge_{p=1}^{j_\ell} \mathbf{F}\,\beta_{\ell,p}$ (thus satisfying β) but with $i \neq l$, so no ψ_k would be satisfied.)

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 $\mathcal{M}^i$,

- $\mathcal{M}^{i+1}$ is built as an extension of $\mathcal{M}^i$ s.t. for each $k < k_i$ $\mathcal{M}^{i+1}, k \models \exists y \cdot \psi[y]$,
- 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 $i = 0$, $k_0 = 0$, $\mathcal{M}^0$ 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 $i > 0$, we 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 Lemma 3, 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 4, 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 $|Const| + 2 \times (K_\psi + 1)$. Then any structure extending $\mathcal{M}^\infty$ is a model of $\mathbf{G}(\exists y \cdot \psi[y])$ of size $|Const| + 2 \times (K_\psi + 1)$. □

4.2. Relaxing the Use of Quantifiers

Our next result, Theorem 2, generalizes the previous result to formulas:

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

The goal of this section is to prove Theorem 2. It can be proved by reduction to Theorem 1. However this requires some preliminary definitions given in the following section.

4.2.1. Definitions

Definition 15. *We say that a formula ϕ is 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 $\phi = \exists y_1 \ldots y_n \cdot \psi$.*

Let $\vec{y} = (y_1, \ldots, y_n)$ be a tuple of distinct variables and $\vec{a} = (a_1, \ldots, a_n)$ a tuple of terms: we denote by $t[\vec{y} \mapsto \vec{a}]$ the parallel substitution of each y_i by each a_i.

Definition 16. *Let $\mathcal{V} = \{y_1, \ldots, y_n\}$ and $\psi \in LTL_{\Sigma,\mathcal{V}}$. Let $\vec{a} = (a_1, \ldots a_n)$ be a tuple of constant symbols. Then we define $\mathcal{T}_\psi(\vec{a}) = \{t[\vec{y} \mapsto \vec{a}] \mid t \in \mathcal{T}_\psi \setminus \mathcal{T}_{\Sigma,\varnothing}\}$.*

Reducing the result of Theorem 2 to Theorem 1 requires us to encode any formula of Gur($\mathbf{X}, \mathbf{F}$) in Gur$^-$($\mathbf{X}, \mathbf{F}$). The following definition formalizes such an encoding.

Definition 17. *Consider a tuple of variables $\vec{y} = (y_1, \ldots, y_n)$. Given an FOLTL formula $\psi[\vec{y}]$ in $LTL_{\Sigma,\{y_1,\ldots,y_n\}}(\mathbf{X}, \mathbf{F})$, we define $\psi^{\vec{y}}$ inductively as follows :*

- $P(t_1, \ldots, t_n)^{\vec{y}} = P_{t_1,\ldots,t_n}(y)$ *where $P_{t_1,\ldots,t_n}$ is a fresh predicate symbol not occuring in ψ;*
- $(O\psi)^{\vec{y}} = O(\psi^{\vec{y}})$ *where $O \in \{\neg, \mathbf{X}, \mathbf{F}\}$.*
- $(\psi O \phi)^{\vec{y}} = \psi^{\vec{y}} O \phi^{\vec{y}}$ *where $O \in \{\vee, \wedge\}$.*

To apply Theorem 1 it is necessary to give a way to build a model of the encoded formula $\mathbf{G}(\exists y \cdot \psi^{\vec{y}}[y])$ from a model of the original formula $\mathbf{G}(\exists \vec{y} \cdot \psi[\vec{y}])$. This construction is given below.

Definition 18. *Given a tuple $\vec{y} = (y_1, \dots, y_n)$, a formula $\psi[\vec{y}]$ and a structure $\mathcal{M}$, we define $\mathcal{M}^{\vec{y}} = (\mathcal{D}^{\vec{y}}, \sigma^{\vec{y}}, \rho^{\vec{y}})$ as follows :*

- $\mathcal{D}^{\vec{y}} = \mathcal{D}^n$;
- $\sigma^{\vec{y}}$ *is the empty function since there is no function symbol to interpret in* $\psi^{\vec{y}}$;
- $P_{t_1,\dots,t_n} \in \rho_i^{\vec{y}}(a_1, \dots, a_n) \Leftrightarrow P \in \rho_i(\sigma(t_1[\vec{y} \mapsto \vec{a}]), \dots, \sigma(t_n[\vec{y} \mapsto \vec{a}]))$.

Then $\mathcal{M}, k, [\vec{y} \mapsto \vec{a}_k] \vDash_p P(t_1, \dots, t_n)$ iff $\mathcal{M}^{\vec{y}}, k, [y \mapsto a_k] \vDash_p P_{t_1,\dots,t_n}(y)$.

From Definition 12, if a partial structure is already defined, and once D_0 is defined, the definition of a new embedded partial structure follows from a given partial embedding function almost immediately. Defining a partial embedding function is simpler than defining an entire partial structure, therefore the previous property allows us to simplify some proofs by defining partial embedding functions instead of some structures. The following lemmas formalize this notion and ensure that it is only necessary to define a domain and a partial embedding function to define a partial structure.

Lemma 6. *Let $\mathcal{M} = (D, \sigma, \rho)$ be a partial structure, D_0 be a set and $f : \mathbb{N} \times D_0 \nrightarrow D$ be a partial function. Then there exists $\mathcal{M}_0 = (D_0, \sigma_0, \rho^0)$ s.t. $\mathcal{M}_0 \overset{f}{\hookrightarrow} \mathcal{M}$.*

Lemma 7. *Let $\mathcal{M}$, $\mathcal{M}_0 = (D_0, \sigma_0, \rho^0)$ and $\mathcal{M}_1 = (D_0, \sigma_1, \rho^1)$ be partial structures and $f : \mathbb{N} \times D_0 \nrightarrow D$ be a partial function. Then if $\mathcal{M}_0 \overset{f}{\hookrightarrow} \mathcal{M}$ and $\mathcal{M}_1 \overset{f}{\hookrightarrow} \mathcal{M}$, we have $\rho^0 = \rho^1$ and, for each $g \in \mathcal{F}_n$, $\vec{d} \in D_0^n$, and each $i \in \mathbb{N}$, if $f_i(\vec{d}) \neq \perp$ then $\sigma_0(g)(\vec{d}) = \sigma_1(g)(\vec{d})$.*

4.2.2. Result

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

Proof. Consider a formula $\phi = \exists \vec{y} \cdot \psi[\vec{y}]$ of Gur$(\mathbf{X}, \mathbf{F})$ and $\mathcal{M}$ a model of $\mathbf{G}(\phi)$. Building a finite model of $\mathbf{G}(\phi)$ can be done by using Theorem 1. In order to apply this theorem, it is necessary to encode ϕ in a formula of Gur$^-(\mathbf{X}, \mathbf{F})$. The previously-defined encoding can be used, which yields the formula $\exists y \cdot \psi^{\vec{y}}[y]$. Then $\mathcal{M}^{\vec{y}}$ defines a model of $\mathbf{G}(\exists y \cdot \psi^{\vec{y}}[y])$. It is now possible to apply Theorem 1 to $\mathcal{M}^{\vec{y}}$ and $\mathbf{G}(\exists y \cdot \psi^{\vec{y}}[y])$. The result of this operation is a structure $\mathcal{M}^{\vec{y},0}$ which is a finite model of $\mathbf{G}(\exists y \cdot \psi^{\vec{y}}[y])$. Then a finite model of $\mathbf{G}(\exists \vec{y} \cdot \psi[\vec{y}])$ can be built from $\mathcal{M}^{\vec{y},0}$. This can be done by somehow reversing the transformation made in Definition 18. First, notice that the partial embedding defined by Theorem 1, denoted as $f^{\vec{y}}$, maps, at each instant, some $x \in \mathcal{D}^{\vec{y},0}$ to a tuple $(y_1, \dots, y_n) \in \mathcal{D}^n$. So we want to define a domain $\mathcal{D}'$ containing at least n copies of $\mathcal{D}^{\vec{y},0}$ so that it is possible to define a partial embedding mapping each x_i to the corresponding y_i, where x was mapped to $(y_1, \dots, y_n)$. However it is necessary to interpret terms appearing in ψ in the domain, so $\mathcal{D}'$ is defined as $\mathcal{D}' = (\mathcal{T}_\psi \cap \mathcal{T}_{\Sigma,\varnothing}) \cup \bigcup_{x \in \mathcal{D}^{\vec{y},0}} \mathcal{T}_\psi(\vec{x})$, where $\vec{x}$ is the tuple of $\biguplus_{i=1}^{n} \mathcal{D}^{\vec{y},0}$ s.t. $\vec{x} = (x_1, \dots, x_n)$ where x_i denotes the i-th copy of x. Now that the domain and the partial embedding function, called f', are defined, Lemmas 6 and 7 give a partial structure $\mathcal{M}'$. The ambiguity for the interpretation of terms is solved as we naturally interpret terms as themselves in the domain. It can be seen from the definition of $\mathcal{M}'$ that:

- $\mathcal{M}' \overset{f'}{\hookrightarrow} \mathcal{M}$
- for each $i \in \mathbb{N}$ and any $x \in \mathcal{D}^{\vec{y},0}$, $f'_i(x_1, \dots x_n) = f_i^{\vec{y}}(x)$.

So we conclude that:

$$\mathcal{M}', i, [\vec{y} \mapsto \vec{d}] \vDash_p P(t_1, \dots, t_n) \text{ iff } \mathcal{M}^{\vec{y},0}, i, [y \mapsto a] \vDash_p P_{t_1,\dots,t_n}(y)$$

Then since $\mathcal{M}^{\vec{y},0}$ is a model of $\mathbf{G}(\exists y \cdot \psi^{\vec{y}}[y])$, $\mathcal{M}'$ is a model of $\mathbf{G}(\exists \vec{y} \cdot \psi[\vec{y}])$. □

4.3. The Geneva Fragment

The fragment used in Theorem 2 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 now, using in particular Lemma 5. Theorem 3 states that we can actually extend the fragment used in Theorem 2 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 gets significantly larger.

Definition 19. *Given a formula ϕ in the form given in Lemma 2. Then we write $\beta_\phi = |\{\beta \mid \exists i \cdot \psi_i = \alpha_i \wedge \ldots \wedge \mathbf{F}\beta \wedge \ldots\}|$.*

Definition 20 (Gex fragment)**.** *We call* Gex 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 3. *The Gex fragment enjoys the FDP. If $\psi \wedge \mathbf{G}(\phi)$ is a satisfiable formula in this fragment, it has a model of size $|(\mathcal{T}_\psi \cup \mathcal{T}_\phi) \cap \mathcal{T}_{\Sigma,\varnothing}| + (1 + 2^{\beta_\phi}) \times (K_\phi + 1) \times |\mathcal{T}_\phi \cap \mathcal{T}_{\Sigma,\mathcal{V}}|$.*

Proof sketch. We now briefly present a sketch of the proof to extend Theorem 1. Adapting the proof to Theorem 2 would not cause any additional difficulty. Let us consider a formula ϕ of class Gur$(\mathbf{X}, \mathbf{F})$, $\psi = \exists y_1, \ldots, y_n \cdot \theta[y_1, \ldots, y_n]$, with $\theta \in \mathrm{LTL}_{\Sigma,\{y_1,\ldots,y_n\}}$, and $\mathcal{M}$ a model of $\psi \wedge \mathbf{G}(\phi)$. First, notice that it is possible, up to a Skolemization, to consider that ψ is an LTL formula. In the proof of Theorem 1 the use of Lemma 5 prevents the satisfaction of ψ. Indeed, by using Lemma 5 we "chop" the first instants of the model. To prove this theorem it is necessary to build a structure where the set of instants of $\mathcal{M}'$ corresponds one to one to instants of $\mathcal{M}$.

The first step is to extend the domain, in Theorem 1 we have $\mathcal{D} = Const \cup \mathcal{D}_\mathbf{X} \cup \mathcal{D}_\mathbf{F}$. Without any loss of generality, let us assume that $\phi = \exists y \cdot \delta$ where δ is in DNF as described in Lemma 2. In that case, $\delta = \psi_1 \vee \ldots \vee \psi_n$ 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 $\alpha_i = \mathbf{X}^{n_{i,1}}\, \ell_{i,1} \wedge \ldots \wedge \mathbf{X}^{n_{i,k_i}}\, \ell_{i,k_i}$;
- each ψ_i is in NNF.

So we define $\mathcal{D}' = \mathcal{D} \uplus (\biguplus_{i=1}^{n} \mathcal{D}_\mathbf{X})$. Let $\mathcal{D}_i$ denote the i-th copy of $\mathcal{D}_\mathbf{X}$. The following steps define the rest of the structure:

- There is an instant s.t. each clause that is satisfied is infinitely often satisfied, after this point we can use the construction of Theorem 1.
- Before this point, for any ψ_i s.t. $\exists y \cdot \psi_i[y]$ is not infinitely often satisfied, there exists some integer $last(i)$ which corresponds to the greatest integer k s.t. $\mathcal{M}, k \vDash_p (\exists y \cdot \psi_i[y])$.
- Before reaching $last(i)$, Lemma 2 can be used as in the proof of Theorem 1 to define the partial embedding over $\mathcal{D}_i$. This ensures that, if $\mathcal{M}, k \vDash_p \exists y \cdot \psi_i[y]$, there is always some $d \in \mathcal{D}_i$ s.t. $\mathcal{M}, k, [y \mapsto d] \vDash_p \alpha_i[y]$.
- Once $last(i)$ is reached (more formally if $k \geq last(i) - K_\phi$) then the partial embedding function is defined so it is "frozen" for the rest of the time. More formally, for each $d \in \mathcal{D}_i$, if k_d denotes the last instant before $last(i)$ s.t. $\mathcal{M}', k_d, [y \mapsto d] \vDash_p \alpha_i[y]$, we define for each $m \geq k_d$, $f'_m(d) = f'_{k_d}(d)$. By construction, we had $\mathcal{M}, k, [y \mapsto f'_{k_d}(d)] \vDash_p \psi_i[y]$. So we ensure that $\mathcal{M}', k, [y \mapsto d] \vDash_p (\beta_{i,1} \wedge \ldots \wedge \beta_{i,j})[y]$.

After these operations, we have $\mathcal{M}', 0 \vDash_p \mathbf{G}(\exists y \cdot \phi[y])$. There is a one-to-one correspondence between instants of $\mathcal{M}'$ and instants of $\mathcal{M}$. Moreover, the interpretation of predicates over closed terms is the same in both structures. Therefore, for any $\delta \in \mathrm{LTL}_{\Sigma,\varnothing}$, we have $\mathcal{M}, 0 \vDash_p \delta$ iff $\mathcal{M}', 0 \vDash_p \delta$. □

Let FO(∀) denote the fragment of purely universal FO formulas containing no other function symbols than constants. The next theorem extends Theorem 3 by allowing formulas of FO(∀) as leaves of the formula instead of basic predicates. However non-constant function symbols cannot 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 21. *An FOLTL formula ψ is in FOLTL$(\exists\uparrow, \forall\downarrow)$ if $\psi = \exists y_1 \ldots y_2 \cdot \theta[y_1, \ldots, y_n]$, 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 6. *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 22. *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 23 (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 4. *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* $|(\mathcal{T}_\psi \cup \mathcal{T}_\phi) \cap \mathcal{T}_{\Sigma,\varnothing}| + (1 + 2^{\beta_\phi}) \times (K_\phi + 1) \times |\mathcal{T}_\phi \cap \mathcal{T}_{\Sigma,\mathcal{V}}|$.

Proof. Given ϕ a formula of FOLTL$(\mathbf{X}, \mathbf{F}, \forall\downarrow)$, ψ a formula of FOLTL$(\exists\uparrow, \forall\downarrow)$ and $\mathcal{M}$ a model of $\psi \wedge \mathbf{G}(\phi)$, it is possible to build ϕ' (resp. ψ') from ϕ (resp. ψ) by replacing any subformula $\delta[y_1, \ldots, y_n] \in \mathrm{FO}(\forall)$ with a predicate $P_\delta(y_1, \ldots, y_n)$. Notice that ϕ' is a formula of Gur$(\mathbf{X}, \mathbf{F})$ and $\psi' = \exists y_1, \ldots, y_n \cdot \theta'[y_1, \ldots, y_n]$, where $\theta' \in \mathrm{LTL}_{\Sigma,\{y_1,\ldots,y_n\}}$. Then it is possible to build a model $\mathcal{M}^0$ of $\psi' \wedge \mathbf{G}(\phi')$ by defining $\mathcal{M}^0$ from $\mathcal{M}$, where the new predicate $P_\delta(y_1, \ldots, y_n)$ holds true iff $\delta[y_1, \ldots, y_n]$ does. Then Theorem 3 can be applied, giving us a finite model $\mathcal{M}'$ of $\psi' \wedge \mathbf{G}(\phi')$ s.t. $\mathcal{M}' \overset{f}{\hookrightarrow} \mathcal{M}$. Then the partial embedding allows to deduce that if $\mathcal{M}, k, f_k \circ C \vDash_p \delta$, we have $\mathcal{M}', k, C \vDash_p \delta$. Then an induction over the structure of the formulas leads to the conclusion that $\mathcal{M}'$ is a model of $\psi \wedge \mathbf{G}(\phi)$. □

4.4. 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. 2.2, this could be a source of infinity axioms if universal quantification is allowed. We show that we can add equality to the $\forall$-free fragments of our previous theorems 1, 2, and 3 and still enjoy the BDP. However, the bound on the domain becomes much larger and not exact anymore.

Definition 24. *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. $\mathrm{LTL}^=_{\Sigma,\mathcal{V}}$) denotes Gur$(\mathbf{X}, \mathbf{F})$ (resp. $\mathrm{LTL}_{\Sigma,\mathcal{V}}$) augmented with equality. Theorem 5 (resp. 6) generalizes Theorem 2 (resp. 3).

Theorem 5. *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 a model $\mathcal{M}$ of $\mathbf{G}(\phi)$ (with $\phi = \exists \vec{y} \cdot \psi$). Theorem 2 can be applied to this formula after replacing equality tests by $\top$. This operation yields a partial structure, which 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 $\mathcal{M}_0$. Remark that this domain is a 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 $\mathcal{M}_0$;
- 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 6. *If ϕ is a formula of class $Gur^{=}(\mathbf{X}, \mathbf{F})$ and $\psi = \exists y_1, \ldots, y_n \cdot \theta[y_1, \ldots, y_n]$, where $\theta \in LTL^{=}_{\Sigma,\{y_1,\ldots,y_n\}}$, 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 5 can easily be adapted to Theorem 6. □

Notice that if we extend the fragment of Theorem 4 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 4 by adding equality remains open.

5. Many-Sorted FOLTL

In this section, we study the BDP for Many-Sorted FOLTL and ultimately present a Many-Sorted extension of the Geneva fragment that enjoys the BDP.

5.1. Many-sorted logic

The definition of Many-Sorted FOLTL requires to define a Many-Sorted signature. Such a signature is defined out of a set $\mathcal{S}$ of sorts.

Definition 25 (Many-sorted signature)**.** *A Many-Sorted signature Σ is a tuple $(\mathcal{S}, \mathcal{R}, \mathcal{F})$ where $\mathcal{S}$ is a set of sorts and:*

- *$\mathcal{R}$ is a family of relation symbols such that if $A = A_1 \ldots A_n \in \mathcal{S}^\star$ then $\mathcal{R}_A$ is the set of relation symbols r over $A_1 \times \ldots \times A_n$.*
- *$\mathcal{F}$ is a family of function symbols such that if $A = A_1 \ldots A_n \in \mathcal{S}^\star$ and $B \in \mathcal{S}$ then $\mathcal{F}_{A,B}$ is the set of function symbols $f : A_1 \times \ldots \times A_n \to B$.*

The definitions of structures and satisfaction relation for mono-sorted logic can be easily extended to Many-Sorted logic. Many-Sorted structures have a set of disjoint non-empty domains, one for each sort. Additionally, the interpretation of relation and function symbols is consistent with their types.

5.2. Stratified fragment of FO

A well-known fragment of FO enjoying the bounded domain property is the Ramsey fragment and its many-sorted extension [13]. In this section we present this fragment that we call the "stratified FO" fragment.

5.2.1. Sort graph

The definition of the stratified fragment requires the definition of the sort graph of a formula. In this graph, an edge from A to B means that adding a constant symbol of sort A *increases* the size of the Herbrand domain corresponding to B. It is then necessary to limit the creation of terms to ensure that this set is finite. The acyclicity of the sort graph is a necessary and sufficient condition for this.

Given a formula ψ, this graph is defined over the set of sorts as follows: there is an edge from A to B iff there is a function symbol $f : A_1 \times ... \times A \times ... \times A_n \to B \in \mathcal{F}$ or there is an existential quantifier of type B in the scope of a universal quantifier of type A in ψ.

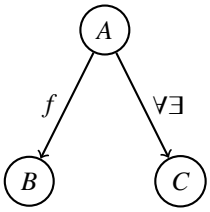
Definition 26 (Sort graph)**.** *Given a formula ψ in NNF, we define $\mathcal{G}_\phi = (\mathcal{S}, E(\phi))$ as follows. For any sorts A, B of $\mathcal{S}$, $(A, B) \in E(\phi)$ if at least one of the following conditions holds:*

- *there exists $\vec{A} = A_1 \ldots A \ldots A_n \in \mathcal{S}^\star$ and $f \in \Sigma_{\vec{A},B}$ s.t. $f \in \mathrm{sub}(\phi)$.*
- *there exists ψ_1, ψ_2 s.t.:*
  - $\forall x : A \cdot \psi_1 \in \mathrm{sub}(\phi)$

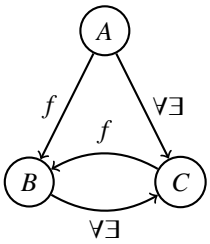
– $\exists y : B \cdot \psi_2 \in \text{sub}(\psi_1)$

Example 3. *In the following examples, we label edges to indicate why they are present (nested quantifiers or function symbol).*

- *Let $f : A \to B$ be a function symbol and $\phi = \exists x : B \cdot \forall y : A \cdot \exists z : C \cdot P(x, y, z, f(y))$ be a formula. Then the sort graph of ϕ is the following:*



- *Let $f : C \times A \to B$ be a function symbol and $\phi = \forall x : B \cdot \forall y : A \cdot \exists z : C \cdot P(x, y, f(z, y))$ be a formula. Then the sort graph of ϕ is the following:*



Lemma 8 (Invariance by Skolemization)**.** *The sort graph of a formula is invariant by Skolemization, in other words we have $\mathcal{G}_\phi = \mathcal{G}_{\phi_{sk}}$.*

Definition 27 (Stratified formula)**.** *A formula $\phi \in FO$ is said to be stratified if $\mathcal{G}_\phi$ is acyclic.*

Theorem 7 (Stratified fragment)**.** *The set of statified FO formulas enjoys the BDP [13].*

5.3. More axioms of infinity

5.3.1. Axiom of infinity with equality relation

Contrary to the unsorted case, adding equality to the naive multi-sorted extension of Theorem 6 leads to axioms of infinity. We now present such an axiom. The main idea is to axiomatize natural numbers indirectly by using equality and a function symbol to specify a successor relation which remains constant through time.

Consider two function symbols $i, s : \mathcal{S}_1 \to \mathcal{S}_2$. Then we define $succ(x, y) := s(x) = i(y)$. Now the rest of the formula will only make intervene the sort $\mathcal{S}_1$, so we will implicitly quantify over this sort for the rest of the example. Besides we assume that we have a constant symbol 0 in the vocabulary of the formula. We start with our first axiom:

$$\forall x, y, z \cdot succ(x, z) \wedge succ(y, z) \Rightarrow x = y \qquad (\phi_{inj})$$

This axiom allows us to show by a simple induction that if we have a formula such that $succ(0, x_1) \wedge ... \wedge succ(x_{j-1}, x_j)$, where there is some non-negative integer $i < j$ such that $x_i = x_j$, then we have $x_{j-i} = 0$. This implies that $succ(x_{j-i-1}, 0)$ holds. Then we add the following axiom to ensure that all x_i are distinct:

$$\forall x \cdot \neg succ(x, 0) \qquad (\phi_{\neg loop})$$

Now to ensure that the axiom implies the existence of an infinite sequence of distinct elements it is sufficient for $succ(0, x_1) \wedge ... \wedge succ(x_{j-1}, x_j)$ to holds for any finite prefix of an infinite sequence $(x_n)_{n \geq 1}$. It can be done with the following axiom:

$$\mathbf{G}(\exists y_1, y_2 P(y_1) \wedge succ(y_1, y_2) \wedge \mathbf{X}(P(y_2) \wedge \forall x \cdot P(x) \Rightarrow x = y_2)) \qquad (\phi_\infty)$$

At each instant the element assigned to y_1 needs to satisfy P and has a successor x_2 assigned to y_2. Then at the next instant x_2 will be the only element to satisfy P, meaning that to satisfy the formula at the next instant it is necessary that x_2 is assigned to y_1, forcing the existence of its successor x_3 to be assigned to y_2. Then this successor will itself have to be assigned to y_1 for the same reasons. This forces the existence of an infinite chain of successors. If this sequence start from 0 then loops are avoided and it ensures an infinite domain. For this purpose the following axiom is added:

$$P(0) \wedge (\forall x \cdot P(x) \Rightarrow x = 0) \qquad (\phi_0)$$

This shows that $\phi_{inj} \wedge \phi_{\neg loop} \wedge \phi_\infty \wedge \phi_0$ is an axiom of infinity.

This example allows us to conclude that using equality in this many-sorted fragment leads to axioms of infinity. For this reason, we prohibit equality in the following section.

5.3.2. Another example of axiom of infinity

As seen in section 2.2, in the unsorted case the stratification condition is not sufficient to ensure the BDP.

A stronger condition is required. For example, we could forbid existential and universal quantifications over the same sort. However, assuming that there is a function symbol $f : A \to B$, the following axiom of infinity can still be exhibited, without quantifying over the same sort:

$$\mathbf{G}(\exists y : A \cdot \neg P(f(y)) \wedge \mathbf{X}\, P(f(y)) \wedge \forall x : B \cdot P(x) \Rightarrow \mathbf{X}\, P(x))$$

5.4. The many-sorted Geneva fragment

Section 5.3.2 shows that the FO stratification condition is not sufficient to ensure the BDP for FOLTL.

Therefore we define an *augmented* sort graph to strengthen the stratification condition. This new sort graph is composed of all the edges of the standard one, as well as new edges depending on the conditions of the following definition.

First, similar to the notation introduced in Sect. 2.1, we let FOLTL($\mathbf{X}, \mathbf{F}, \mathbf{G}$) denote the set of FOLTL formulas in NNF where the set of temporal operators used in the formula is restricted to $\{\mathbf{X}, \mathbf{F}, \mathbf{G}\}$.

Definition 28 (Sort graph)**.** *Given a formula $\phi \in$ FOLTL($\mathbf{X}, \mathbf{F}, \mathbf{G}$), we define the labeled* sort graph *of ϕ, $\mathcal{G}_\phi = (\mathcal{S}, E(\phi))$, as follows. For any sorts A, B of $\mathcal{S}$, if at least one of the following conditions holds, then $(A, \ell, B) \in E(\phi)$:*

1. *There exists $\vec{A} = A_1 \dots A \dots A_n \in \mathcal{S}^\star$ and $f \in \Sigma_{\vec{A},B}$ s.t. $f \in \mathrm{sub}(\phi)$.*
   And then the label is defined as $\ell = f$.
2. *$A = B$ and there exists ψ_1, ψ_2, ψ_3 s.t.:*
   - $\mathbf{G}\,\psi_1 \in \mathrm{sub}(\phi)$
   - $\exists y : A \cdot \psi_2 \in \mathrm{sub}(\psi_1)$
   - $\mathbf{G}\,\psi_3 \in \mathrm{sub}(\psi_2)$

   And then the label is defined as $\ell = \mathbf{G}\,\exists\,\mathbf{G}$.
3. *There exists ψ_1, ψ_2, ψ_3 s.t.:*
   - $\mathbf{G}\,\psi_1 \in \mathrm{sub}(\phi)$
   - $\forall x : A \cdot \psi_3 \in \mathrm{sub}(\phi)$ *and ψ_3 contains a temporal connective*
   - $\exists y : B \cdot \psi_2 \in \mathrm{sub}(\psi_1) \cap \mathrm{sub}(\psi_3)$

   And then the label is defined as $\ell = \forall\,\mathbf{G}\,\exists$.
4. *The conditions of* (2) *are not fulfilled and there exists ψ_1, ψ_2 s.t.:*
   - $\forall x : A \cdot \psi_1 \in \mathrm{sub}(\phi)$
   - $\exists y : B \cdot \psi_2 \in \mathrm{sub}(\psi_1)$

   And then the label is defined as $\ell = \forall\exists$.
5. *There exists ψ_1, ψ_2, ψ_3 s.t.:*
   - $\mathbf{G}\,\psi_1 \in \mathrm{sub}(\phi)$

- $\exists y : B \cdot \psi_2 \in \text{sub}(\psi_1)$
- $\forall x : A \cdot \psi_3 \in \text{sub}(\phi)$ *and* ψ_3 *contains a temporal connective*

And then the label is defined as $\ell = \mathbf{G}\,\exists\forall$.

Remark 7. *In Def. 28, the definition of labels is unnecessary to determine if a formula enjoys the BDP but is needed to compute the corresponding bound.*

Example 4. *Here are some examples of formulas illustrating the definition of the sort graph.*

- *The first example is one of the axioms of infinity presented in 2.2.*

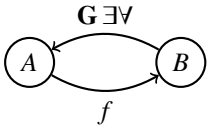
$$\phi = \mathbf{G}(\exists y : A \cdot \neg P(y) \wedge \mathbf{X}\,P(y) \wedge \forall x : A \cdot P(x) \Rightarrow \mathbf{X}\,P(x))$$

Then the sort graph of ϕ *is the following:*

- *The following axiom of infinity was introduced in Sect. 5.3.2. Given a function symbol* $f : A \to B$*:*

$$\phi = \mathbf{G}(\exists y : A \cdot \neg P(f(y)) \wedge \mathbf{X}\,P(f(y)) \wedge \forall x : B \cdot P(x) \Rightarrow \mathbf{X}\,P(x))$$

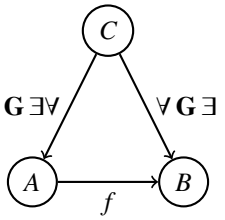
Then the sort graph of ϕ *is the following:*



- *Let* $f : A \to B$ *be a function symbol. Given the formula:*

$$\begin{aligned}\phi \quad = \quad & \mathbf{G}\Big[\exists y : A \cdot \neg P(f(y)) \wedge \mathbf{X}\,P(f(y)) \wedge \\ & \forall x : C \cdot \exists z : B \cdot Q(x,z) \Rightarrow \mathbf{X}(\forall w : A \cdot P(w))\Big]\end{aligned}$$

the sort graph of ϕ *is the following:*



Definition 29 (Temporal stratification)**.** *A formula* $\phi \in FOLTL(\mathbf{X}, \mathbf{F}, \mathbf{G})$ *is called temporally stratified iff* $\mathcal{G}_\phi$ *is acyclic. The fragment of temporally stratified formula of* $FOLTL(\mathbf{X}, \mathbf{F}, \mathbf{G})$ *is called MS-Geneva*

Theorem 8. *Any formula* $\phi \in$ *MS-Geneva enjoys the BDP. The computation of the corresponding bound is given by Algorithm 1.*

Algorithm 1 Computation of the bound for ϕ

Require: $\phi \in \mathrm{FOLTL}(\mathbf{X}, \mathbf{F}, \mathbf{G}) \wedge \phi$ Skolemized $\wedge(\mathcal{S}, E) = \mathcal{G}_\phi \wedge \mathcal{G}_\phi$ acyclic
Ensure: $\forall S \in \mathcal{S}, N_S$ is the bound for the domain corresponding to the sort S

for $S \in \mathcal{S}$ **do**
  $N_S := |Const_S| + (1 + 2^{\beta_\phi}) \times (K_\phi + 1) \times |\mathcal{V}_S|$
while $E \neq \varnothing$ **do**
  for $S \in \mathcal{S}$ **do**
    if $\forall S' \in \mathcal{S}, S' \to S \notin E$ **then**
      for $f \in \mathcal{F} \wedge \exists S' \in \mathcal{S} \cdot \{S \xrightarrow{f} S'\} \in E$ **do**
        $A := 1$
        for $S' \in \mathcal{S} \wedge S \xrightarrow{f} S' \in E$ **do**
          $E := E \setminus \{S \xrightarrow{f} S'\}$
          $A := A \times N_{S'}$
        $N_{S'} := N_{S'} + A$
      for $S' \in \mathcal{S} \wedge S \xrightarrow{\forall \mathbf{G} \exists} S' \in E$ **do**
        $E := E \setminus \{S \xrightarrow{\forall \mathbf{G} \exists} S'\}$
        $N_{S'} := N_{S'} \times N_S$
      for $S' \in \mathcal{S} \wedge S \xrightarrow{\mathbf{G} \exists \forall} S' \in E$ **do**
        $E := E \setminus \{S \xrightarrow{\mathbf{G} \exists \forall} S'\}$

Proof. In a first step, we show that w.l.o.g., we can consider a temporally stratified formula ϕ such that any existential quantifier appears in the scope of a a $\mathbf{G}$ operator. Indeed, a trivial generalization to FOLTL of Lemma 8 implies that the sort graph is invariant by Skolemization. Therefore, it is possible to transform any temporally stratified formula in Skolemized form. Such a form exactly satisfies the condition above.

Then we consider $\phi \in \mathrm{FOLTL}_\Sigma(\mathbf{X}, \mathbf{F}, \mathbf{G})$ such that:

- ϕ is satisfiable
- If $\exists y : B \cdot \psi \in \mathrm{sub}(\phi)$ then there is some formula ψ' s.t. $\exists y : B \cdot \psi \in \mathrm{sub}(\mathbf{G}\,\psi')$ and $\mathbf{G}\,\psi' \in \mathrm{sub}(\phi)$.

The only feature that prevents ϕ from belonging to the Geneva fragment is the possible presence of universal quantifiers over non temporal formulas. In the following, we show that, since the formula is temporally stratified, it is possible to unfold each of these universal quantifiers into a finite conjunction over the set of closed terms of the corresponding sort. This unfolding operation, written $\_^\exists$, is defined as follows:

- if $\psi \in \mathrm{FO}$ then $\psi^\exists = \psi$.
- $(\forall x : A \cdot \psi(x))^\exists = \bigwedge_{t \in \mathcal{T}_{\Sigma,A}} \psi(t)^\exists$.
- If $O \in \{\vee, \wedge\}$ then $(\psi_1 O \psi_2)^\exists = (\psi_1^\exists) O (\psi_2^\exists)$, if $O \in \{\mathbf{X}, \mathbf{F}, \mathbf{G}\}$ then $(O\psi)^\exists = O(\psi^\exists)$ and if $A \in \mathcal{S}$ then $(\exists y : A \cdot \psi)^\exists = \exists y : A \cdot (\psi^\exists)$.

Consider now a structure $\mathcal{M}$ such that the domain of each sort is equal to the set of closed terms of this sort. Then it is easy to show that $\mathcal{M} \vDash_p \phi$ iff $\mathcal{M} \vDash_p \phi^\exists$. We must therefore exhibit such a finite model for $\phi^\exists$, which will yield a finite model for ϕ.

However some additional transformations are required in order to apply Theorem 4 to $\phi^\exists$. Indeed, Theorem 4 requires the formula to be of the shape $\alpha \wedge \mathbf{G}\beta$ with the following conditions:

- α is an existentially quantified LTL combination of universally quantified propositional formulas (see Def. 21);
- β is an existentially quantified formula formed by combining universally quantified propositional formulas with $\mathbf{X}, \mathbf{F}, \wedge, \vee$ connectives (see Def. 22);

So, we now a formula of the form $\alpha \wedge \mathbf{G}\beta$ as defined above that is equisatisfiable with $\phi^{\exists}$[2]. First, we define α as the result of $\boldsymbol{\alpha}(\phi^{\exists})$, where $\boldsymbol{\alpha}(-)$ is a transformation defined by induction over subformulas of $\phi^{\exists}$ in the following way:

- if $\gamma = \ell$ is a literal then $\boldsymbol{\alpha}(\gamma) = \ell$
- if $\gamma = \exists y : B \cdot \gamma_1$ then :
  - if there exists $\gamma_2 \in$ FOLTL s.t. $\mathbf{G}\,\gamma_2 \in \text{sub}(\phi^{\exists})$ and $\gamma \in \text{sub}(\gamma_2)$[3] then $\boldsymbol{\alpha}(\gamma) = P_\gamma$ (where P_γ denotes a fresh predicate)
  - otherwise, $\boldsymbol{\alpha}(\gamma) = \exists y : B \cdot \boldsymbol{\alpha}(\gamma_1)$
- if $\gamma = O\gamma_1$, where $O \in \{\mathbf{X}, \mathbf{F}, \mathbf{G}\}$, then $\boldsymbol{\alpha}(\gamma) = O\boldsymbol{\alpha}(\gamma_1)$.
- if $\gamma = \gamma_1 O \gamma_2$, where $O \in \{\wedge, \vee\}$, then $\boldsymbol{\alpha}(\gamma) = \boldsymbol{\alpha}(\gamma_1) O \boldsymbol{\alpha}(\gamma_2)$.

Thus, $\boldsymbol{\alpha}(\phi^{\exists})$ fits into the left term of the Geneva fragment.

Second, writing Γ for the set of all formulas of the shape $\exists y : B \cdot \gamma_1$ that have been replaced by a fresh predicate while applying $\boldsymbol{\alpha}(-)$), we define $\beta = \bigwedge_{\alpha \in \Gamma} P_\gamma \Rightarrow \gamma$.

Ultimately, $\boldsymbol{\alpha}(\phi^{\exists}) \wedge \mathbf{G}(\beta)$ enjoys the following properties:

- it fits into the Geneva fragment;
- it is equisatisfiable with $\phi^{\exists}$;
- all its models are models of $\phi^{\exists}$ ($\alpha \wedge \mathbf{G}\beta \vDash \phi^{\exists}$);
- it is satisfiable because ϕ is assumed to be satisfiable.

For these reasons, we can consider a finite model $\mathcal{M}^{\exists}$ of $\phi^{\exists}$ obtained by Theorem 4 applied to $\boldsymbol{\alpha}(\phi^{\exists}) \wedge \mathbf{G}(\beta)$. We remark that the only "sources for elements" of the domain of $\mathcal{M}^{\exists}$ are:

- closed terms,
- terms built over elements added to satisfy existential quantifiers nested under $\mathbf{G}$ operators.

Therefore, if $D_{\mathcal{M}^{\exists},B} \neq \sigma_{\mathcal{M}^{\exists}}(\mathcal{T}_{\Sigma,B})$, there is a path from some sort A to B where A is existentially quantified under a $\mathbf{G}$ operator. However, by definition of the sort graph, there is a $\mathbf{G}\,\exists\forall$ edge from A to B in the sort graph, implying a cycle in the sort graph and contradicting the fact that ϕ is temporally stratified. Then we conclude that $D_{\mathcal{M}^{\exists},B} = \sigma_{\mathcal{M}^{\exists}}(\mathcal{T}_{\Sigma,B})$.

This leads to the fact that $\mathcal{M}^{\exists}, 0 \vDash_p \phi^{\exists}$ iff $\mathcal{M}^{\exists}, 0 \vDash_p \phi$. Since $\mathcal{M}^{\exists}$ is a domain-finite model of $\phi^{\exists}$, it is a domain-finite model of ϕ. So ϕ enjoys the BDP.

We now prove that Algorithm 1 computes a correct bound for the considered formula. We proceed by induction over the number of existential quantifiers in the scope of a universal quantifier.

- If there is no existential quantifier in the scope of a universal quantifier then unfolding universal quantifiers does not create any additional element in the bounded domain used for Theorem 4. Then the bound given by the algorithm corresponds to the bounds from Theorem 4.
- If there is an existential quantifier over sort A in the scope of a universal quantifier over sort B but not under a $\mathbf{G}$ operator, then it is possible to Skolemize this quantifier in the classical way. This operation implies to add the bound of the domain of B to the bound for A. The bound for B is necessarily computable by acyclicity of the sort graph.
- If there is an existential quantifier over sort A in the scope of a universal quantifier over sort B and under a $\mathbf{G}$ operator, then it is necessary to unfold the universal quantifier. This operation imposes to duplicate the existential quantifier over A as many times as the number of elements in the domain of B. In the worst case, it may multiply the bound of the sort B by the bound of the sort A. □

[2] In practice, this proof not only shows the equisatisfiability of the two formulas, but also that $\alpha \wedge \mathbf{G}\beta \vDash \phi^{\exists}$.

[3] The set of all formulas $\gamma = \exists y : B \cdot \gamma_1 \in \text{sub}(\phi^{\exists})$ satisfying this condition is called Γ

6. MS-Geneva at work

In this section, we present an illustrative example in the form of a simple distributed protocol. The properties verified in this example are not designed to be hard to check using classical tools. Rather, they were chosen to demonstrate some advantages and limitations appearing when trying to apply our results to real distributed protocols. We also show that it is possible to abstract the specification of the system to overcome these difficulties. Notice *en passant* that the said abstractions are designed to enjoy automation, which will be the topic of a future article

The example we present is a notification system in a ring-shaped network[4]. This protocol features a sort $\mathtt{N}$ for nodes, a sort $\mathtt{M}$ for messages, a predicate $\mathtt{succ}$ relating a node to its successor and a predicate $\mathtt{rcvd}$ representing the fact that a given node have received a message.

A formula **Ring** (unshown here to save space) specifies that $\mathtt{succ}$ forms a finite ring. As exemplified in Padon *et al.* [4], a finite ring can be axiomatized in pure FO. It can then be used to describe a *static* ring in FOLTL (such a description does not fit in MS-Geneva, but more on that later).

The rest of the specification says that any node that has received some message may send it to its successor, while other nodes do not change during this operation.

$$\begin{array}{lcl}
\mathbf{Same}(z,m) & := & \mathtt{rcvd}(z,m) \Leftrightarrow \mathbf{X}\,\mathtt{rcvd}(z,m) \\
\mathbf{Send}(x,m) & := & \mathtt{rcvd}(x,m) \wedge \exists y : \mathtt{N} \left[\mathtt{succ}(x,y) \wedge \mathbf{X}\,\mathtt{rcvd}(y,m) \right. \\
 & & \left. \quad \wedge (\forall z : \mathtt{N}, m' : \mathtt{M} \cdot (z \neq y \vee m \neq m') \Rightarrow \mathbf{Same}(z,m')) \right] \\
\mathbf{Trans} & := & \mathbf{G}(\exists p : \mathtt{N}, m : \mathtt{M} \cdot \mathbf{Send}(p,m))
\end{array}$$

6.1. Safety Property

Now consider the safety property "if a node is notified of a message, it remains notified of this message", described by the following formula: $\mathbf{Safety} := \mathbf{G}(\forall x : \mathtt{N}, m : \mathtt{M} \cdot \mathtt{rcvd}(x,m) \Rightarrow \mathbf{X}\,\mathtt{rcvd}(x,m))$. 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.

Note that $\neg\mathbf{Safety} \equiv \exists x : \mathtt{N}, m : \mathtt{M} \cdot \mathbf{F}(\mathtt{rcvd}(x,m) \wedge \mathbf{X}\,\neg\mathtt{rcvd}(x,m))$, therefore an equi-satisfiable formula can be obtained by Skolemization:

$$\mathbf{SkNegSafety} := \mathbf{F}(\mathtt{rcvd}(c,d) \wedge \mathbf{X}\,\neg\mathtt{rcvd}(c,d))$$

However $\varphi := \mathbf{Ring} \wedge \mathbf{Trans} \wedge \mathbf{SkNegSafety}$ is *not* in any of our fragments because of the universal quantification over $\mathbf{Same}(z,m')$, 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 4.

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 this, we instantiate the variable z with the values x and c, and the variable m' with the values m and d, which yields:

$$\begin{array}{l}
\overline{\mathbf{Send}}(x,m) := \mathtt{rcvd}(x,m) \\
\qquad\qquad \wedge \exists y : \mathtt{N} \cdot (\mathtt{succ}(x,y) \wedge \mathbf{X}\,\mathtt{rcvd}(y,m) \\
\qquad\qquad\qquad \wedge (c \approx y \vee m \approx d \vee \mathbf{Same}(z,m')) \wedge (c \approx y \Leftrightarrow \mathbf{X}\,c \approx y))
\end{array}$$

Notice that 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 $c \approx y$ and $\neg\,\mathbf{X}\,c \approx y$. In this case, no constraint would apply to the truth value of $\mathbf{X}\,\mathtt{notified}(c)$.

Thus, it is possible to define an abstraction by the following formulas: $\overline{\mathbf{Trans}} := \mathbf{G}(\exists p : \mathtt{N}, m : \mathtt{M} \cdot \overline{\mathbf{Send}}(p,m))$ and $\mathbf{AbsSatS} := \overline{\mathbf{Eq}} \wedge \overline{\mathbf{Trans}} \wedge \mathbf{SkNegSafety}$. Remark that **Ring** does not intervene in **AbsSatS** so even if the specification of a static ring does not fit into MS-Geneva, it does not cause harm here.

It is easy to show that $\mathbf{Ring} \wedge \mathbf{Trans} \wedge \neg\mathbf{Safety} \models \mathbf{AbsSatS}$ and that **AbsSatS** belongs to the fragment of Theorem 8. Applying this theorem, we compute a size of 5 for nodes and a size of 3 for messages. Using the Electrum tool, **AbsSatS** can be shown to be unsatisfiable for these bounds, which ultimately proves the original property.

[4] We provide a corresponding Electrum specification, available at `https://gitlab.com/grayswandyr/notificationspec/-/tree/master/IC2020`. The Electrum tool can be obtained at `http://haslab.github.io/Electrum`.

6.2. Liveness Property

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

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, using the Skolem constant, the universal quantifiers that are out of our fragment.

This time, an axiom *abstracting* the ring topology must be added in order to prove the liveness property. This axiom states that if all messages in the ring at some instant are eventually *transferred* along the `succ` relation, then any message in the ring in the initial state will eventually propagate to any node of the ring. Formally, the FOLTL formula describing this axiom is:

$$\forall n : M \cdot \Big[((\exists x : N \cdot \mathtt{rcvd}(x, n)) \wedge \mathbf{G}\,\mathbf{Transfer}) \Rightarrow \forall x : N \cdot \mathbf{F}\,\mathtt{rcvd}(x, n)\Big]$$

with the following *transfer* formula:

$$\mathbf{Transfer} := \forall x : N, m : M \cdot \mathtt{rcvd}(x, m) \Rightarrow [\exists y : N \cdot \mathbf{F}(\mathtt{succ}(x, y) \wedge \mathtt{rcvd}(y, m))].$$

Remark that this abstraction fits within a more general framework than presented in this paper which will be the subject of future work.

In the end, the obtained formula fits into the fragment of Theorem 8, which provides a size of 6 for the domain corresponding to nodes and a size of 3 for the domain corresponding to messages. The formula can be shown in Electrum to be unsatisfiable for a bound of 6, which proves the property.

7. Related Work

In [10], 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 **X** and **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 **G**. Notice this fragment is strictly included in MS-Geneva.

The decidable monodic fragment studied by Hodkinson *et al.* [14, 15] 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.* [4] 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 [5, 6] 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.

8. Conclusion

In the introduction, we drew as an inspiration for our work the following classical shape for specifications of systems and of their properties: $spec = init \wedge \mathbf{G}\,trans \wedge fair \rightarrow prop$ (with *trans* using only the **X** connective). Checking the validity of *spec* amounts to assessing the satisfiability of $\neg spec = init \wedge \mathbf{G}\,trans \wedge fair \wedge \neg prop$. Our results then say that this satisfiability can be decided provided $\neg spec$ respects the conditions of Theorems 3, 4, 6 or 8.

In the spririt of [4, 13, 16, 17], we devised a many-sorted version of the Geneva fragment presented in [12], which extends its expressiveness and fits the data structuring features of Electrum [7]. Also and contrary to [12] there is no

need to split and "dispatch" *fair* and *prop* anymore. Theorem 8 allows for more flexibility in the use of existential and universal quantifiers.

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 **F** connectives under a **G**. This ultimately yields a decision procedure for the validity of *spec*. Notice that existing tools, such as our own Electrum [7], can readily be used to support it, as was shown in Sect. 6.

A 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. 6. However the stratification condition prevents for any sort A, to existentially quantify over A under a **G** connective and universally quantify over A over a temporal formula. Depending on the problem at hand, this can be restrictive.

In the future, we will address some current limitations of our fragments. Indeed, some statements like fairness assumptions or frame conditions (which specify what does *not* change when a transition happens) cannot always be expressed naturally in our fragments. The main approach we will investigate is the study of different mechanisms of abstraction that could be used to transform a protocol described in MSFOLTL into an abstract protocol that fits into MS-Geneva. We will also assess our approach on more realistic case studies.

On a more theoretical side, we will also study the computational complexity of our fragments. We can already build from our previous work [10]. Indeed, we studied in that article the complexity of the satisfiability problem of (full) FOLTL for bounded-domain models (taking the domain bound as an input). Then, for our fragments, which enjoy the BDP, we could easily deduce an upper bound of the complexity of the satisfiability problem from the expression of the bound on the domain.

Acknowledgements

We thank the anonymous reviewers as well as Prof. Jean-Paul Bodeveix for their helpful comments.

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.

References

[1] F. Kröger, S. Merz, Temporal Logic and State Systems (Texts in Theoretical Computer Science. An EATCS Series), Springer, 2008.

[2] D. M. Gabbay, A. Kurucz, F. Wolter, M. Zakharyaschev, Many-Dimensional Modal Logics: Theory and Applications, Elsevier, 2003, Ch. Fragments of first-order temporal logics.

[3] E. Börger, E. Grädel, Y. Gurevich, The Classical Decision Problem, Perspectives in Mathematical Logic, Springer, 1997. doi:10.1007/978-3-642-59207-2.

[4] O. Padon, K. L. McMillan, A. Panda, M. Sagiv, S. Shoham, Ivy: safety verification by interactive generalization, ACM SIGPLAN Notices 51 (6) (2016) 614–630. doi:10.1145/2980983.2908118.

[5] O. Padon, J. Hoenicke, G. Losa, A. Podelski, M. Sagiv, S. Shoham, Reducing liveness to safety in first-order logic, Proceedings of the ACM Conference on Principles of Programming Languages (POPL) 2 (2017) 26. doi:10.1145/3158114.

[6] O. Padon, J. Hoenicke, K. L. McMillan, A. Podelski, M. Sagiv, S. 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.

[7] N. Macedo, J. Brunel, D. Chemouil, A. Cunha, D. Kuperberg, Lightweight Specification and Analysis of Dynamic Systems with Rich Configurations, in: Foundations of Software Engineering, Seattle, United States, 2016. doi:10.1145/2950290.2950318.

[8] J. Brunel, D. Chemouil, A. Cunha, N. Macedo, The Electrum Analyzer: Model Checking Relational First-Order Temporal Specifications, in: 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE '18), ACM Press, Montpellier, France, 2018. doi:10.1145/3238147.3240475.

[9] J. Brunel, D. Chemouil, J. Tawa, Analyzing the fundamental liveness property of the Chord protocol, in: Formal Methods in Computer Aided Design (FMCAD), 2018, pp. 1–9. doi:10.23919/FMCAD.2018.8603001.

[10] D. Kuperberg, J. Brunel, D. Chemouil, On finite domains in first-order linear temporal logic, in: Automated Technology for Verification and Analysis, Springer, 2016, pp. 211–226. doi:10.1007/978-3-319-46520-3_14.

[11] L. Lamport, Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers, Addison-Wesley Professional, 2002.

[12] Q. Peyras, J. Brunel, D. Chemouil, A bounded domain property for an expressive fragment of first-order linear temporal logic, in: 26th International Symposium on Temporal Representation and Reasoning (TIME 2019), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[13] A. Abadi, A. Rabinovich, M. Sagiv, Decidable fragments of many-sorted logic, Journal of Symbolic Computation 45 (2) (2010) 153–172. doi:10.1016/j.jsc.2009.03.003.

[14] I. Hodkinson, F. Wolter, M. Zakharyaschev, Decidable fragments of first-order temporal logics, Annals of Pure and Applied logic 106 (1-3) (2000) 85–134. doi:10.1016/S0168-0072(00)00018-X.

[15] I. Hodkinson, F. Wolter, M. Zakharyaschev, Monodic fragments of first-order temporal logics: 2000–2001 a.d., in: Logic for Programming, Artificial Intelligence, and Reasoning, Springer, 2001, pp. 1–23. doi:10.1007/3-540-45653-8_1.

[16] T. Nelson, D. J. Dougherty, K. Fisler, S. Krishnamurthi, On the finite model property in order-sorted logic, Tech. rep., Worcester Polytechnic Institute (2010).

[17] T. Nelson, D. J. Dougherty, K. Fisler, S. Krishnamurthi, Toward a more complete Alloy, in: Abstract State Machines, Alloy, B, VDM, and Z, Springer, 2012, pp. 136–149. doi:10.1007/978-3-642-30885-7_10.