UNIVERSITÀ DEGLI STUDI DI MILANO

Dipartimento di Scienze dell'Informazione

RAPPORTO INTERNO N° 313-07

Combination Methods for Satisfiability and Model-Checking of Infinite-State Systems

Silvio Ghilardi, Enrica Nicolini, Silvio Ranise, Daniele Zucchelli

Combination Methods for Satisfiability and Model-Checking of Infinite-State Systems

Silvio Ghilardi[1], Enrica Nicolini[2], Silvio Ranise[2], and Daniele Zucchelli[1,2]
[1]Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano (Italy)
[2]LORIA & INRIA-Lorraine

March 1, 2007

Abstract

Manna and Pnueli have extensively shown how a mixture of first-order logic (FOL) and discrete Linear time Temporal Logic (LTL) is sufficient to precisely state verification problems for the vast class of reactive systems. Theories in FOL model the (possibly infinite) data structures used by a reactive system while LTL specifies its (dynamic) behavior. The combination of LTL and FOL allows us to specify infinite state systems and the subtle ways in which their data flow influences the control flow. Indeed, the capability of automatically solving satisfiability and model-checking problems is of paramount importance to support the automation of verification techniques using this framework.

In this paper, we derive undecidability and decidability results for both the satisfiability of (quantifier-free) formulae and the model-checking of safety properties by lifting combination methods for (non-disjoint) theories in FOL. The proofs of our decidability results suggest how decision procedures for the constraint satisfiability problem of theories in FOL and algorithms for checking the satisfiability of propositional LTL formulae can be integrated. This paves the way to employ efficient Satisfiability Modulo Theories solvers in the model-checking of infinite state systems, as previous proposals have suggested their use for bounded model-checking. We exemplify our techniques on some examples.

Contents

1 Introduction

In [33] and many other writings, Manna and Pnueli have extensively shown how a mixture of first-order logic (FOL) and discrete Linear time Temporal Logic (LTL) is sufficient to precisely state verification problems for the class of reactive systems. Theories in FOL model the (possibly infinite) data structures used by a reactive system while LTL specifies its (dynamic) behavior. The combination of LTL and FOL allows one to specify infinite state systems and the subtle ways in which their data flow influences the control flow. Indeed, the capability of automatically solving satisfiability and model-checking problems is of paramount importance to support the automation of verification techniques using this framework.

A lot of efforts addressed both the satisfiability and the model-checking problem of Propositional LTL. The former has been attacked using a range of techniques from tableaux [26] to extensions of resolution [30]. More recently, extensions of resolutions have also been used to solve the satisfiability problem of the Monodic fragment of First-Order LTL (see, e.g., [31]). Techniques based on automata [29] or symbolic methods (see, e.g., [10]) have been put forward to solve the model-checking problem of finite-state systems. Significant work has been done also in the context of model-checking for infinite-state systems (see, e.g., [46, 39, 7] to name but a few). An integration of classic tableaux and automated deduction techniques is presented in [40]. The so-called "abstract-check-refine" techniques for model-checking of infinite state systems combine finite-state model checking and decision procedures for first-order theories (see, e.g., [27, 28]). Also, bounded model-checking of infinite state systems based on the use of Satisfiability Modulo Theories (SMT) solvers have been investigated more recently (see, e.g., [13]).

We briefly describe here our framework for *integrating LTL operators with theories in FOL* (see Section 2.4 for more): we fix a theory T in a first-order signature Σ and consider as a temporal model a succession $\mathcal{M}_1, \mathcal{M}_2, \ldots$ of ordinary models of T, provided such models share the same carrier (otherwise said, the domain of the temporal model is 'constant'). Following [38], we also declare symbols from a subsignature Σ_r of Σ to be *rigid*: this means that in a temporal model $\mathcal{M}_1, \mathcal{M}_2, \ldots$ the Σ_r-restrictions of the $\mathcal{M}_i$'s must coincide (free variables are similarly divided into 'rigid' and 'flexible' ones). For model-checking purposes, first-order *initial and transition formulae* are specified, whose role is that of (non-deterministically) restricting the temporal model evolution (see Section 4 for details).

We derive undecidability and decidability results for both the satisfiability of *quantifier-free* formulae and the model-checking of safety properties by lifting combination methods for (non-disjoint) theories in FOL. As pointed out by Manna and Pnueli in [33], although the restriction to quantifier-free formulae decreases the expressive power of the logic, such a class

of formulae is sufficient for many verification problems.

The *first contribution* of the paper is a reduction of the satisfiability problem for quantifier-free LTL formulae modulo the background theory T to an instance of the Nelson-Oppen combination problem for first-order theories (the combination being disjoint if the rigid subsignature is empty). More precisely, we consider a theory T whose constraint satisfiability problem consists of non-deterministically solving one of the (decidable) constraint satisfiability problem of two signature-disjoint theories T_1, T_2. It is not difficult to see that the constraint satisfiability problem of T is decidable. Although the satisfiability problem of T is decidable, it is possible to write a quantifier-free LTL formula whose satisfiability is equivalent to the satisfiability of a constraint in $T_1 \cup T_2$, which turns out to be undecidable, if T_1 and T_2 are chosen as shown in [5]. The undecidability of the safety model-checking problem follows (under mild hypotheses) by a well-known reduction to the reachability problem for Minsky machines [35].

Since the satisfiability problem for quantifier-free LTL formulae modulo a background theory T looks very much like a non-disjoint combination problem, the hope is that the same (or similar) requirements yielding the decidability of the constraint satisfiability problem in unions of theories [23], will also give decidability here. The *second contribution* of the paper is to show that this is indeed the case: we have decidability of the satisfiability problem for quantifier-free LTL formulae modulo T, in case T has decidable universal fragment and is T_r-compatible, where T_r is the restriction of the universal fragment of T to the rigid subsignature (for termination, one must also assume T_r to be locally finite or noetherian).

The *third* (and main) *contribution* of the paper is that (under the same T_r-compatibility and local finiteness hypotheses) the model-checking problem for quantifier-free safety properties is also decidable. The proof of this result suggests how decision procedures for the constraint satisfiability problem of theories in FOL and algorithms for checking the satisfiability of propositional LTL formulae can be integrated. This paves the way to employ *efficient Satisfiability Modulo Theories solvers* in the model-checking of infinite state systems, as previous proposals have suggested their use for bounded model-checking. We exemplify our techniques on some examples.

Plan of the paper. Section 2 introduces the background notions on first-order logic, combination methods for non-disjoint theories, propositional and first-order (quantifier-free) temporal logic. Section 3 and Section 4 give the undecidability and decidability results for the satisfiability problem of quantifier-free formulae of first-order temporal logic and the model-checking problem of safety properties, respectively. Section 5 discusses related work and Section 6 concludes the paper with a discussion of the achieved results and some future work.

The Appendix contains some more background materials and technical results on non-disjoint combination.

2 Formal Preliminaries

2.1 First-Order Logic

A *signature* Σ is a set of functions and predicate symbols (each endowed with the corresponding arity). We assume the binary equality predicate symbol '=' to be always present in any signature Σ (so, if $\Sigma = \emptyset$, then Σ does not contain other symbols than equality). To avoid confusion, we use the symbol $\equiv$ (instead of $=$) in the metalanguage to mean identity of syntactic expressions. The signature obtained from Σ by adding a set $\underline{a}$ of new constants (i.e., 0-ary function symbols) is denoted by $\Sigma^{\underline{a}}$. Σ-*terms*, Σ-*atoms*, Σ-*literals*, Σ-*clauses*, and (elementary) Σ-*formulae* are defined in the usual way (we will omit the prefix Σ when it is clear from the context). A *positive clause* is a disjunction of atoms. A *constraint* is a conjunctions of literals. Terms, literals, clauses and formulae are called *ground* whenever no variable appears in them. Formulae without free variables are *sentences*. A Σ-*theory* T is a set of sentences (called the axioms of T) in the signature Σ. A formula is *quantifier-free* (or open) iff it does not contain quantifiers. The universal (resp., existential) closure of a formula φ is the sentence obtained by adding to φ a prefix of universal (resp., existential) quantifiers binding all variables which happen to have a free occurrence in φ.

Below, letters $\varphi, \psi, \ldots$ are used for formulae; the following notations will be used below: $\varphi(\underline{x})$ means that the set of free variables in φ is contained in the finite set $\underline{x}$ whereas $\varphi(\underline{a}/\underline{x})$ (or, simply, $\varphi(\underline{a})$ leaving the $\underline{x}$ implicit) means that $\varphi(\underline{a})$ is the formula obtained from $\varphi(\underline{x})$ by the componentwise replacement of the free variables in $\underline{x}$ by the constants in $\underline{a}$.

From the semantic side, we have the standard notion of a Σ-*structure* $\mathcal{M} = (M, \mathcal{I})$: this is nothing but a support set M endowed with an arity-matching interpretation $\mathcal{I}$ of the function and predicate symbols from Σ. We use $f^{\mathcal{M}}$ (resp. $P^{\mathcal{M}}$) to denote the interpretation of the function symbol f (resp. predicate symbol P) in the structure $\mathcal{M}$ (the equality predicate $=$ is always interpreted as the identity relation over M). *Truth* of a Σ-formula in $\mathcal{M}$ is defined in any one of the standard ways (so that truth of a formula is equivalent to truth of its *universal* closure). We let $\bot$ denote an arbitrary formula which is true in no structure. A formula φ is *satisfiable* in $\mathcal{M}$ iff its *existential* closure is true in $\mathcal{M}$.

A Σ-structure $\mathcal{M}$ is a *model* of a Σ-theory T (in symbols $\mathcal{M} \models T$) iff all the sentences of T are true in $\mathcal{M}$. If φ is a formula, $T \models \varphi$ ('φ *is a logical consequence of* T') means that φ is true in all the models of T ($T \models \varphi$ is equivalent to $T \models \forall \underline{x}\, \varphi$, where $\forall \underline{x}\, \varphi$ is the universal closure of φ). A Σ-theory T is *complete* iff for every Σ-sentence φ, either φ or $\neg\varphi$ is a logical

consequence of T; T is *consistent* iff it has a model, i.e., $T \not\models \bot$. A sentence φ is T-consistent iff $T \cup \{\varphi\}$ is consistent.

A theory is *universal* iff it has universal closures of open formulae as axioms. A Σ-theory T admits *quantifier elimination* iff for every formula $\varphi(\underline{x})$ there is a quantifier-free formula $\varphi'(\underline{x})$ such that $T \models \varphi(\underline{x}) \leftrightarrow \varphi'(\underline{x})$. There are many well-known theories eliminating quantifiers [9]; e.g., Linear (Integer[1] or Rational) Arithmetics, Real Arithmetics, acyclic lists, and any theory axiomatizing enumerated datatypes.

The *constraint satisfiability problem* for the constraint theory T is the problem of deciding whether a Σ-constraint is satisfiable in a model of T (or, equivalently, T-satisfiable).[2] In the following, we use free constants instead of variables in constraint satisfiability problems, so that we (equivalently) redefine a constraint satisfiability problem for the theory T as the problem of *establishing the consistency of* $T \cup \Gamma$ *for a finite set* Γ *of ground* $\Sigma^{\underline{a}}$*-literals* (where $\underline{a}$ is a finite set of new constants). For the same reason, from now on, *by a '*Σ*-constraint' we mean a 'ground* $\Sigma(\underline{a})$*-constraint'*, where the finite set of free constants $\underline{a}$ should be clear from the context (if not explicitly mentioned).

If $\Sigma_0 \subseteq \Sigma$ is a subsignature of Σ and if $\mathcal{M}$ is a Σ-structure, the Σ_0*-reduct* of $\mathcal{M}$ is the Σ_0-structure $\mathcal{M}_{|\Sigma_0}$ obtained from $\mathcal{M}$ by forgetting the interpretation of function and predicate symbols from $\Sigma \setminus \Sigma_0$.

A Σ*-embedding* (or, simply, an embedding) between two Σ-structures $\mathcal{M} = (M, \mathcal{I})$ and $\mathcal{N} = (N, \mathcal{J})$ is any mapping $\mu : M \longrightarrow N$ among the corresponding support sets satisfying the condition

$$\mathcal{M} \models \varphi \quad \text{iff} \quad \mathcal{N} \models \varphi \tag{1}$$

for all Σ^M-atoms φ (here $\mathcal{M}$ is regarded as a Σ^M-structure, by interpreting each additional constant $a \in M$ into itself and $\mathcal{N}$ is regarded as a Σ^M-structure by interpreting each additional constant $a \in M$ into $\mu(a)$). Notice the following facts: (a) as we have equality in the language, an embedding is an injective function; (b) an embedding $\mu : \mathcal{M} \longrightarrow \mathcal{N}$ must be an algebraic homomorphism, that is for every n-ary function symbol f and for every $a_1, \ldots, a_n \in M$, we must have $f^{\mathcal{N}}(\mu(a_1), \ldots, \mu(a_n)) = \mu(f^{\mathcal{M}}(a_1, \ldots, a_n))$;[3] (c) for an n-ary predicate symbol P we must have $(a_1, \ldots, a_n) \in P^{\mathcal{M}}$ iff $(\mu(a_1), \ldots, \mu(a_n)) \in P^{\mathcal{N}}$. It is easily seen that an embedding $\mu : \mathcal{M} \longrightarrow \mathcal{N}$ can be equivalently defined as a mapping $\mu : M \longrightarrow N$ satisfying (a)-(b)-(c) above.

[1] For integer arithmetic, infinite predicates expressing equivalence modulo n must be included in the language in order for quantifiers to be eliminable.

[2] Notice that the complementary constraint unsatisfiability problem (i.e. the problem of deciding whether a finite set of Σ-literals is unsatisfiable in all the models of T) is easily reduced to the problem of deciding whether $T \models \varphi$ holds, for quantifier-free φ.

[3] To see this, apply (1) to the Σ^M-atom $f(a_1, \ldots, a_n) = a$, where $a \in M$ is just $f^{\mathcal{M}}(a_1, \ldots, a_n)$.

If $M \subseteq N$ and if the embedding $\mu : \mathcal{M} \longrightarrow \mathcal{N}$ is just the identity inclusion $M \subseteq N$, we say that $\mathcal{M}$ is a *substructure* of $\mathcal{N}$ or that $\mathcal{N}$ is an *extension* of $\mathcal{M}$. In case (1) holds for all first order formulae, the embedding μ is said to be an *elementary* embedding. Correspondingly, in case μ is also an inclusion, we say that $\mathcal{M}$ is an elementary substructure of $\mathcal{N}$ or that $\mathcal{N}$ is an elementary extension of $\mathcal{M}$.

2.2 Background on Combination

We recall some notions used to develop results [23, 25, 2, 1, 24] for the non-disjoint combination of theories. This paper is self-contained, in the sense that proofs of all model-theoretic facts which are needed for our results on temporal logic are fully given in Appendix A. However, we refer the reader to [23] for more information and for the proofs of side claims we are making in this section (these side claims will never be used within the paper, but might be useful for a better insight into the notions we are going to introduce).

2.2.1 Compatible Theories

Definition 2.1 (T_0-compatibility [23])**.** Let T be a theory in the signature Σ and let T_0 be a universal theory in a subsignature $\Sigma_0 \subseteq \Sigma$. We say that T is T_0-*compatible* iff $T_0 \subseteq T$ and there is a Σ_0-theory T_0^* such that

(i) $T_0 \subseteq T_0^\star$;

(ii) $T_0^\star$ has quantifier elimination;

(iii) every model of T_0 can be embedded into a model of $T_0^\star$;

(iv) every model of T can be embedded into a model of $T \cup T_0^\star$.

The requirements (i)-(iii) make the theory $T_0^\star$ unique, provided it exists ($T_0^\star$ is nothing but the so called *model completion* of T_0 [9]).[4]

In principle, we do not need to have a characterization of $T_0^\star$, the mere information of its existence is enough for our decision procedures to be sound and complete and to implement them. As for T_0 itself, it is usually sufficient to take as T_0 the set of universal Σ_0-sentences which are logical consequence of T (for instance, this will be always the case for the temporal logic decision problems analyzed in this paper). No information will be needed on axiomatizations of T_0 to run our decision procedures too, we shall just need qualitative information on properties of T_0, like local finiteness, noetherianity, etc. (see below).

[4]The standard definition of model completion (adopted also in [23]) is slightly different, but can be proved to be equivalent to the above one in the case of universal theories, see the Appendix B of [22] for details.

A lot of examples of theories fitting Definition 2.1 can be easily obtained as follows: suppose that T_0^* is a Σ_0-theory that eliminates quantifiers and take T be any theory *whatsoever* in a bigger signature such that $T \supseteq T_0^*$. Then T is T_0-compatible, if we take as T_0 the theory having as axioms all the universal Σ_0-sentences which are logical consequences of $T_0^\star$.

Of course, the key requirements in Definition 2.1 are requirements (iii)-(iv). Such requirements trivialize in the case considered in the last paragraph; to understand what they mean, notice that (by Robinson diagram theorem and by compactness) they are equivalent to the following statements:

(iii') *every Σ_0-constraint which is satisfiable in a model of T_0 is satisfiable also in a model of $T_0^\star$*;[5]

(iv') *every Σ-constraint which is satisfiable in a model of T is satisfiable also in a model of $T_0^\star \cup T$.*[6]

These requirements are nothing but a generalization of the stable infiniteness requirement of the Nelson-Oppen combination procedure [36], [44]: in fact, if T_0 is the empty theory in the empty signature, $T_0^\star$ is the theory axiomatizing an infinite domain, so that (iii') holds trivially and (iv') is precisely stable infiniteness.

Other examples of T_0-compatible theories are given in [23]: for instance, any extension (in a richer functional signature and by means of equational axioms) of the theory BA of Boolean algebras is BA-compatible.

2.2.2 Locally Finite and Noetherian Theories

T_0-compatibility is used in order to obtain the completeness of combination algorithms; for termination, local finiteness and noetherianity are the relevant requirements.

Definition 2.2 (Local Finiteness)**.** We say that Σ_0-theory T_0 is *locally finite* iff Σ_0 is finite and, for every finite set of free constants $\underline{a}$, there are finitely many ground $\Sigma_0^{\underline{a}}$-terms $t_1, \ldots, t_{k_{\underline{a}}}$ such that for every further ground $\Sigma_0^{\underline{a}}$-term u, we have that $T_0 \models u = t_i$ (for some $i \in \{1, \ldots, k_{\underline{a}}\}$). If such $t_1, \ldots, t_{k_{\underline{a}}}$ are effectively computable from $\underline{a}$, then T_0 is said to be *effectively locally finite*.

If Σ_0 is finite and does not contain any function symbol, then any Σ_0-theory is effectively locally finite; among effectively locally finite theories we have Boolean algebras, linear arithmetic modulo a fixed integer, and theories axiomatizing enumerated datatypes.

[5] Equivalently, T_0 and $T_0^\star$ entail the same universal Σ_0-sentences.

[6] Equivalently, T and $T \cup T_0^\star$ entail the same universal Σ-sentences.

The main way in which local finiteness is exploited lies in the computation of finite *representatives* sets of ground atoms, clauses and formulae[7] in finitely expanded signatures. This means the following (e.g. in the case of atoms): consider the signature $\Sigma_0^{\underline{a}}$, obtained from Σ_0 by expanding it with finitely many free constants $\underline{a}$. Thanks to effective local finiteness of T_0, it is possible to compute finitely many $\Sigma_0^{\underline{a}}$-atoms $\psi_1(\underline{a}), \ldots, \psi_m(\underline{a})$ such that for any further $\Sigma_0^{\underline{a}}$-atom $\psi(\underline{a})$ there is some i such that $T_0 \models \psi_i(\underline{a}) \leftrightarrow \psi(\underline{a})$. These atoms $\psi_1(\underline{a}), \ldots, \psi_m(\underline{a})$ are called representatives (modulo T_0-equivalence) because they can freely replace arbitrary $\Sigma_0^{\underline{a}}$-atoms in computational considerations.

Local finiteness is a quite strong requirement: in many cases a much weaker requirement is sufficient. This requirement is called a 'noetherianity' requirement, because it generalizes standard conditions from abstract algebra.

Definition 2.3 (Noetherian Theory)**.** A Σ_0-theory T_0 is *noetherian* if and only if for every *finite* set of constants $\underline{a}$, every infinite ascending chain

$$\Theta_1 \subseteq \Theta_2 \subseteq \cdots \subseteq \Theta_n \subseteq \cdots$$

of sets of ground $\Sigma_0^{\underline{a}}$-atoms is eventually constant modulo T_0 (meaning that there is an n such that for all m and $A \in \Theta_m$, we have $T_0 \cup \Theta_n \models A$).

Examples of noetherian theories are linear integer and linear rational arithmetic (provided ordering is not included in the language). The reason why rational linear arithmetic is noetherian is simple and is due to the fact that there cannot be infinitely many linearly independent linear equations in finitely many unknowns (notice that this theory is far from being locally finite, though).

A new interesting example of a noetherian theory which is presented in Appendix A is *the empty theory over the signature* Σ *containing only a unary function symbol*. Notice that if we add further axioms to a noetherian theory (while keeping the signature fixed), we still result in a noetherian theory:[8] thank to this observation, the theory of on injective function, or of a cycle-free unary function, etc. are easily seen to be noetherian.

For the last three definitions below, we fix two signatures $\Sigma_0 \subseteq \Sigma$, a Σ_0-theory T_0, and a Σ-theory T such that $T_0 \subseteq T$. Moreover, in this context it is useful to *include the inconsistent proposition* $\perp$ *among atoms*.

Definition 2.4 (T_0-basis)**.** Given a finite set of ground Σ-clauses Θ and a finite set of free constants $\underline{a}$, a T_0*-basis* for Θ w.r.t. $\underline{a}$ is a set Δ of positive ground $\Sigma_0^{\underline{a}}$-clauses such that:

[7] Recall that when we say that a formula is ground we mean that it does not contain variables, neither free nor bounded.

[8] The same observation applies to the property of being effectively locally finite.

(i) $T \cup \Theta \models C$ for all $C \in \Delta$;

(ii) if $T \cup \Theta \models C$ then $T_0 \cup \Delta \models C$ for every positive ground $\Sigma_0^{\underline{a}}$-clause C.

We point out that Θ from Definition 2.4 is a finite set of ground Σ-clauses that may contain free constants other from $\underline{a}$; however only the $\underline{a}$ may occur in a T_0-basis for Θ w.r.t. $\underline{a}$.[9]

Definition 2.5 (Noetherian Residue Enumerator)**.** Given a finite set of free constants $\underline{a}$, a *T-residue enumerator for T_0 w.r.t. $\underline{a}$* is a computable function $Res_T^{\underline{a}}(\Theta)$ mapping a finite set of Σ-clauses Θ to a T_0-basis of Θ w.r.t. $\underline{a}$.

An argument based on König lemma (see [24] or Lemma A.11 from Appendix A) shows that if T_0 is a noetherian theory, then for every finite set of Σ-clauses Θ and for every $\underline{a}$, there always exists a finite T_0-basis for Θ w.r.t. $\underline{a}$. However, such a basis is not necessarily computable; to this aim we introduce the following

Definition 2.6. A theory T is an *effectively noetherian extension of* T_0 if and only if T_0 is noetherian and, for every finite set of free constants $\underline{a}$, there is a T-residue enumerator for T_0 w.r.t. $\underline{a}$.

Examples of effectively noetherian extensions of a noetherian theory T_0 can be found in computational algebra (e.g. by using Gröbner bases [24]). The further example given by the pair (T = real linear arithmetic, T_0 = real linear arithmetic without $<$) is considered in [37] (notice that since T_0 eliminates quantifiers, T is not only effectively noetherian in T_0, but also T_0-compatible).

2.2.3 A Combination Schema for Non-Disjoint Theories

In this section, we review the combination results of [23], taking into consideration also further extensions from [24]: these results will not be used in the remaining part of the paper (hence we omit proofs), nevertheless they might be useful in order to understand the role played within combination problems by the notions introduced so far.

Suppose we are given theories T_1, T_2 in signatures Σ_1, Σ_2 and suppose that constraint satisfiability problem is decidable for both T_1 and T_2; *what can we say about constraint satisfiability problem for the $(\Sigma_1 \cup \Sigma_2)$-theory $T_1 \cup T_2$*? In general, not so much: constraint satisfiability problem in $T_1 \cup T_2$ can be undecidable, even if the shared signature $\Sigma_1 \cap \Sigma_2$ is

[9] Notice that, once residue enumerators for Σ-constraints are known, one can get also residue enumerators for finite sets of Σ-clauses (thus, there is no real difference among the definitions we give here and those introduced in [24]).

empty [5]. We look for sufficient conditions making this 'decidability transfer result' available. We first state the following basic combination result from [23]:

Theorem 2.7. *Suppose that the theories T_1, T_2 (in signatures Σ_1, Σ_2) both have decidable constraint satisfiability problem; then the $(\Sigma_1 \cup \Sigma_2)$-theory $T_1 \cup T_2$ also has decidable constraint satisfiability problem in case T_1, T_2 are both T_0-compatible for some universal and effectively locally finite $(\Sigma_1 \cap \Sigma_2)$-theory T_0 contained in T_1, T_2.*

As pointed out in Section 2.2.1, to get concrete applications of Theorem 2.7 it is sufficient to take any theories T_1, T_2 extending a locally finite quantifier eliminating theory $T_0^\star$ in the shared signature $\Sigma_1 \cap \Sigma_2$ (the T_0 fitting the hypotheses of Theorem 2.7 is then the theory whose axioms are all the universal consequences of $T_0^\star$): examples of such a $T_0^\star$ include linear arithmetic modulo n, the theory of dense total orders without endpoints, or any theory axiomatizing enumerated datatypes. Another family of applications (covering the fusion decidability transfer result for global consequence relation in modal logic [47]) arises by taking as T_1, T_2 equational extensions of the theory BA of Boolean algebras (in this case, the hidden $T_0^\star$ is the theory of atomless Boolean algebras, see [23] for details). Finally, it should be clear that Theorem 2.7 extends Nelson-Oppen combination result for disjoint signatures (take T_0 to be the empty theory and $T_0^\star$ to be the theory of an infinite domain).

The algorithm suggested by the plain proof of Theorem 2.7 consists in the following three steps:

Step 1. The input $(\Sigma_1 \cup \Sigma_2)$-constraint Γ is *purified*, in the sense that, by repeatedly adding to it equations like $a = t$ (here t is a term occurring in Γ and a is a fresh constant), an equi-satisfiable constraint $\Gamma_1 \cup \Gamma_2$ is produced, where Γ_i is a Σ_i-constraint for $i = 1, 2$;

Step 2. A maximal $\Sigma_0^{\underline{a}}$-constraint Δ is guessed (here Σ_0 is the shared signature $\Sigma_1 \cap \Sigma_2$, whereas the $\underline{a}$'s are the free constants occurring in both Γ_1 and Γ_2). A $\Sigma_0^{\underline{a}}$-constraint Δ is maximal iff for every $\Sigma_0^{\underline{a}}$-atom ψ, Δ contains a literal which is T_0-equivalent either to ψ or to $\neg\psi$ (notice that maximal constraints are computable, and finitely many modulo T_0, thanks to effective local finiteness of T_0).

Step 3. Return "`Satisfiable`" iff $\Gamma_1 \cup \Delta$ is T_1-satisfiable and $\Gamma_2 \cup \Delta$ is T_2-satisfiable; return "`Unsatisfiable`" iff all guessing Δ fail.

A slightly different proof of Theorem 2.7 suggests an alternative algorithm, based on propagation instead of guessing. Even better, instead of propagating entailed positive clauses (like in [23]), a splitting mechanism with backtracking can be used, as suggested in [24]. To this aim, instead of Steps 2-3, the following Loop is executed after Step 1:

Loop. Pick a positive Σ_0^a-clause C (let C be $A_1 \vee \cdots \vee A_n$, for $n \geq 1$) such that $\Gamma_i \cup \{\neg A_1, \ldots, \neg A_n\}$ is T_i-unsatisfiable but $\Gamma_j \cup \{\neg A_1, \ldots, \neg A_n\}$ is T_j-satisfiable (for $i, j \in \{1, 2\}$, $j \neq i$); choose nondeterministically $k \in \{1, \ldots, n\}$ and update the current constraints by $\Gamma_1 := \Gamma_1 \cup \{A_k\}$ and $\Gamma_2 := \Gamma_2 \cup \{A_k\}$.

Since T_0 is locally finite, there are only finitely many Σ_0^a-positive clauses, hence the Loop cannot be executed forever; when exiting the Loop, the procedure returns "`Satisfiable`" iff $\Gamma_1 \cup \Delta$ is T_1-satisfiable and $\Gamma_2 \cup \Delta$ is T_2-satisfiable, otherwise it backtracks (and returns "`Unsatisfiable`" if backtracking has been completed).[10]

As remarked in [24], the backtracking version of the combined decision algorithm is sound and complete (although not terminating) even in case T_0 lacks the local finiteness requirement: in fact, T_0-compatibility requirements and a fair selection strategy for the positive clause examined in the Loop are sufficient to guarantee completeness. In order to re-gain termination, a noetherianity requirement can be used, witness the following Theorem proved in [24]:

Theorem 2.8. *Suppose that the theories T_1, T_2 (in signatures Σ_1, Σ_2) both have decidable constraint satisfiability problem; then the $(\Sigma_1 \cup \Sigma_2)$-theory $T_1 \cup T_2$ also has decidable constraint satisfiability problem in case there is some universal and noetherian $(\Sigma_1 \cap \Sigma_2)$-theory T_0 such that T_1, T_2 are both T_0-compatible effectively noetherian extensions of T_0.*

The algorithm underlying the proof of Theorem 2.8 is analogous to the backtracking version of the combined decision algorithm in the effectively locally finite case: the main Loop is executed by a fair strategy based on the T_i-residue enumerators for T_0 and termination is guaranteed by noetherianity.

2.3 Propositional Discrete Linear Time Temporal Logic

Propositional $\mathcal{L}$-formulae (or PLTL-formulae or simply propositional formulae) are built up from a set of propositional letters $\mathcal{L}$ by using boolean connectives and the temporal operators $X, \Box, \Diamond, U$. We use letters $\alpha, \beta, \ldots$ for propositional formulae. The semantics for PLTL is the standard one: we recall it for the sake of completeness. A *PLTL-Kripke model $V = \{V_n\}_n$ for $\mathcal{L}$* is a sequence of boolean assignments

$$V_n : \mathcal{L} \longrightarrow \{0, 1\} \qquad (n \in \mathbb{N}).$$

Given such a Kripke model and a propositional formula α, the notion of α being true at instant $t \in \mathbb{N}$ in V is recursively defined as follows (this is parallel to Definition 2.10):

[10] Notice that backtracking is not needed if T_1, T_2 are both Σ_0-convex theories (in the sense of [43]), because in this case we can limit ourselves to positive unit clauses in the Loop.

– if $p \in \mathcal{L}$, $V \models_t p$ iff $V_t(p) = 1$;

– $V \models_t \neg\alpha$ iff $V \not\models_t \alpha$;

– $V \models_t \alpha \wedge \beta$ iff $V \models_t \alpha$ and $V \models_t \beta$;

– $V \models_t \alpha \vee \beta$ iff $V \models_t \alpha$ or $V \models_t \beta$;

– $V \models_t X\alpha$ iff $V \models_{t+1} \alpha$;

– $V \models_t \Box\alpha$ iff for each $t' \geq t$, $V \models_{t'} \alpha$;

– $V \models_t \Diamond\alpha$ iff for some $t' \geq t$, $V \models_{t'} \alpha$;

– $V \models_t \alpha U \beta$ iff there exists $t' \geq t$ such that $V \models_{t'} \beta$ and for each t'', $t \leq t'' < t' \Rightarrow V \models_{t''} \alpha$.

We say that α is satisfied in V iff $V \models_0 \alpha$ (in general, if the subscript of $\models$ is omitted, it is intended to be equal to 0).

2.4 First-Order Discrete Linear Time Temporal Logic

The aim of this paper is that of studying reactive systems descriptions by combining temporal operators and first-order languages. As argued in [33] (p. 48), for most applications it is sufficient to fix a first-order signature Σ and to deal with formulae obtained by applying temporal and boolean operators (but no quantifiers) to first-order Σ-formulae: the resulting formulae are called *state-quantified formulae* in [33] and are formally introduced as follows.

Definition 2.9 (LTL($\Sigma^{\underline{a}}$)-Sentences)**.** Given a signature Σ and a (finite or infinite) set of free constants $\underline{a}$, the set of LTL($\Sigma^{\underline{a}}$)-sentences is inductively defined as follows:

– if φ is an first-order $\Sigma^{\underline{a}}$-sentence, then φ is an LTL($\Sigma^{\underline{a}}$)-sentence;

– if ψ_1, ψ_2 are LTL($\Sigma^{\underline{a}}$)-sentence, so are $\psi_1 \wedge \psi_2$, $\psi_1 \vee \psi_2$, $\neg\psi_1$, $X\psi_1$, $\Box\psi_1$, $\Diamond\psi_1$, $\psi_1 U \psi_2$.

Notice that free constants are allowed in the definition of an LTL($\Sigma^{\underline{a}}$)-sentence. This is quite conventional: since we prefer not to use free variables, free constants handle variables and parameters of the system to be modeled.

Let us now discuss *semantic* issues. It is clear that an LTL($\Sigma^{\underline{a}}$)-structure must be a family of $\Sigma^{\underline{a}}$-structures $\mathcal{M} = \{\mathcal{M}_n = (M_n, \mathcal{I}_n)\}_{n \in \mathbb{N}}$ indexed by the natural numbers; when we fix also a background Σ-theory T, these structures will be taken to be models of T. The main question is the following: what should the various $\mathcal{M}_n$ share? A first requirement is that they should share their domains, that is we assume the M_n to be *constant*, i.e. all equal to each

other. Although different semantics, with increasing and even distinct domains, have been proposed in the literature [6], the constant domain assumption is rather common in computer science applications.

Definition 2.10. Given a signature Σ and a set $\underline{a}$ of free constants, an *LTL($\Sigma^{\underline{a}}$)-structure* (or simply a *structure*) is a sequence $\mathcal{M} = \{\mathcal{M}_n = (M, \mathcal{I}_n)\}_{n\in\mathbb{N}}$ of $\Sigma^{\underline{a}}$-structures. The set M is called the *domain* (or the *universe*) and $\mathcal{I}_n$ is called the *n-th level interpretation function* of the LTL($\Sigma^{\underline{a}}$)-structure.[11]

The mere requirement of domains to be constant is rather poor for our applications, but it is sufficient to formalize the following definition of semantics.

Definition 2.11. Given an LTL($\Sigma^{\underline{a}}$)-sentence φ and $t \in \mathbb{N}$, the notion of "φ being true in the LTL($\Sigma^{\underline{a}}$)-structure $\mathcal{M} = \{\mathcal{M}_n = (M, \mathcal{I}_n)\}_{n\in\mathbb{N}}$ at the instant t" (in symbols $\mathcal{M} \models_t \varphi$) is inductively defined as follows:

- if φ is an first-order sentence, $\mathcal{M} \models_t \varphi$ iff $\mathcal{M}_t \models \varphi$;
- $\mathcal{M} \models_t \neg\varphi$ iff $\mathcal{M} \not\models_t \varphi$;
- $\mathcal{M} \models_t \varphi \wedge \psi$ iff $\mathcal{M} \models_t \varphi$ and $\mathcal{M} \models_t \psi$;
- $\mathcal{M} \models_t \varphi \vee \psi$ iff $\mathcal{M} \models_t \varphi$ or $\mathcal{M} \models_t \psi$;
- $\mathcal{M} \models_t X\varphi$ iff $\mathcal{M} \models_{t+1} \varphi$;
- $\mathcal{M} \models_t \Box\varphi$ iff for each $t' \geq t$, $\mathcal{M} \models_{t'} \varphi$;
- $\mathcal{M} \models_t \Diamond\varphi$ iff for some $t' \geq t$, $\mathcal{M} \models_{t'} \varphi$;
- $\mathcal{M} \models_t \varphi U \psi$ iff there exists $t' \geq t$ such that $\mathcal{M} \models_{t'} \psi$ and for each t'', $t \leq t'' < t' \Rightarrow \mathcal{M} \models_{t''} \varphi$.

The definition above is well given because, if the main connective of the formula is a boolean operator, the definition of truth of an LTL($\Sigma^{\underline{a}}$)-sentence coincides with truth clause of Tarski semantics for first order languages. Let φ be an LTL($\Sigma^{\underline{a}}$)-sentence; φ *is true in* $\mathcal{M}$ or, equivalently, that $\mathcal{M}$ *satisfies* φ (in symbols $\mathcal{M} \models \varphi$) iff $\mathcal{M} \models_0 \varphi$.

[11] In more detail, $\mathcal{I}_n$ is such that $\mathcal{I}_n(P) \subseteq M^k$ for every predicate symbols $P \in \Sigma$ of arity k, and $\mathcal{I}_n(f) : M^k \longrightarrow M$ for each function symbol $f \in \Sigma$ of arity k.

2.4.1 LTL-Theories and the Satisfiability Problem

Let us now better examine the problem of the relationship between the interpretations $\mathcal{I}_n$ in an LTL($\Sigma^{\underline{a}}$)-structure: there are two radically opposite alternatives to cope with this problem. The customary Kripkean semantics for modal logics mostly deals with purely relational signatures and leave the interpretation of the predicate symbols *flexible*, i.e. time-dependant: no relationship among $\mathcal{I}_m(P)$ and $\mathcal{I}_n(P)$ is assumed for $n \neq m$. By contrast, constants are usually interpreted *rigidly* according to the orthodox Kripkean viewpoint, that is we have $\mathcal{I}_m(c) = \mathcal{I}_n(c)$ for all m, n and for all constants c.

On the other hand, the verification literature tends to consider the opposite solution: free constants are flexible (because the system variables are subject to change during runs) and symbols from Σ are rigidly interpreted, because they are supposed to model datatypes endowed with the corresponding time-independent operations (like sum and successor for integers, read/write for arrays, etc.).

While keeping the same motivations of the verification literature, we adopt here a more elaborated point of view, according to which *certain symbols are declared rigid and the remaining ones are declared flexible* (i.e. time-dependent). We believe that there are various reasons supporting this choice. First of all, flexible interpretations are already used within the verification literature, where not only variables, but also propositions expressing program locations are in fact interpreted in a time-dependent way (to this aim, the booleans sort is introduced in order to assimilate program locations to flexible variables). Moreover, reactive systems are supposed to interact with the environment and the environment action is somewhat unpredictable, to the point that it is better to model it through flexible function symbols - these function symbols obeying only to the constraints expressed by the background theory T or by the nondeterministic transition relation of the system (to see an example of what we mean, see the functions *in* and *out* within the water level controller example discussed in Subsection 4.4 below). Even predicates or function symbols expressing the internal evolution of the system may be subject to time change. Consider for instance a mutual exclusion protocol, like the 'bakery' protocol: here the set of processors wanting to enter into the critical section is variable and the ticket-assigning function is time-dependent too, e.g. because it need complete reset once the resource have been obtained (see again Subsection 4.4 for details). In these examples, the *constrained flexibility approach* we propose identifies the good *abstraction* level for the specification of the system behavior. Finally, there are also technical reasons supporting our proposal: big decidability problems in model-checking arise when even minimal infinite states descriptors enter into the picture (see the proof of Theorem 4.10 below) and our setting allows to model the system by grouping problematic descriptors

into two categories, the rigid and the flexible ones. As we shall see, if we succeed in keeping the rigid part of the specification relatively simple (e.g. 'locally finite'), then we do not loose the nice properties of the reasoning about finite-state specifications.

The above discussion leads to the following notions.

Definition 2.12. An *LTL-theory* is a 5-tuple $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ where

- Σ is a signature;
- T is a Σ-theory (called the underlying theory of $\mathcal{T}$);
- Σ_r is a subsignature of Σ;
- $\underline{a}, \underline{c}$ are sets of free constants.

Σ_r is said to be the *rigid subsignature* of the LTL-theory; the constants $\underline{c}$ will be rigidly interpreted, whereas the constants $\underline{a}$ will be interpreted in a time-dependant way. The constants $\underline{a}$ are (slightly improperly) called the *system variables* of the LTL-theory, and the constants $\underline{c}$ are called its *system parameters*.

An LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ is said to be *totally flexible* iff Σ_r is empty and is said to be *totally rigid* iff $\Sigma_r = \Sigma$. Thus, parameters are the only rigid symbols of a totally flexible LTL-theory and system variables are the only flexible symbols of a totally rigid LTL-theory.

Definition 2.13. An LTL($\Sigma^{\underline{a},\underline{c}}$)-structure $\mathcal{M} = \{\mathcal{M}_n = (M, \mathcal{I}_n)\}_{n \in \mathbb{N}}$ is *appropriate* for an LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ iff for all $m, n \in \mathbb{N}$, for all function symbol $f \in \Sigma_r$, for all relational symbol $P \in \Sigma_r$, and for all constant $c \in \underline{c}$, we have

$$\mathcal{M}_n \models T, \quad \mathcal{I}_n(f) = \mathcal{I}_m(f), \quad \mathcal{I}_n(P) = \mathcal{I}_m(P), \quad \mathcal{I}_n(c) = \mathcal{I}_m(c).$$

The *satisfiability problem for* $\mathcal{T}$ is the following: given an LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence φ, decide whether there is an LTL($\Sigma^{\underline{a},\underline{c}}$)-structure $\mathcal{M}$ appropriate for $\mathcal{T}$ such that $\mathcal{M} \models \varphi$. The *ground satisfiability problem for* $\mathcal{T}$ is similarly introduced, but φ is assumed to be ground.

Notice that our definition agrees with the requirement that the equality symbol is always interpreted as the identity relation, because the equality is included in every signature (hence also in the rigid signature Σ_r).

2.4.2 Some Classes of LTL-Theories

To study the ground satisfiability problem for LTL-theories, it is useful to distinguish three different classes of LTL-theories of increasing expressiveness and to lift to the temporal level

the properties of (first-order) theories ensuring modularity (with respect to unions of theories) of decidability of constraint satisfiability problem (cf. Section 2.2.2).

Let Σ be a finite signature; an *enumerated datatype theory* in the signature Σ is the theory consisting of the set of sentences which are true in a finite given fixed Σ-structure $\mathcal{M} = (M, \mathcal{I})$ (we require $\mathcal{M}$ to have the additional property that for every $m \in M$ there is $c \in \Sigma$ such that $c^{\mathcal{M}} = m$). It is easy to see that an enumerated datatype theory has a finite set of universal axioms and enjoys quantifier elimination.

Definition 2.14. An LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ is said to be *finite state* iff it is totally rigid and T is an enumerated datatype theory.

Notice that enumerated datatype theories are locally finite, but not conversely;[12] thus, in order to generalize finite state systems, one can require the underlying theory to be locally finite. We also want to drop the total rigidity requirement and weaken the quantifier elimination property of enumerated datatype theories to a compatibility requirement (recall Definition 2.1):

Definition 2.15. An LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ is said to be *locally finite compatible* iff there is a universal and effectively locally finite Σ_r-theory T_r such that T is T_r-compatible.

Notice that, from our discussion in Section 2.2.1, it follows that a totally flexible LTL-theory is locally finite compatible in case its underlying theory is stably infinite.

We can get a further generalization by weakening local finiteness to noetherianity (in the sense of Definition 2.6):

Definition 2.16. An LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ is said to be *noetherian compatible* iff there is a Σ_r-universal theory T_r such that T is an effectively noetherian and T_r-compatible extension of T_r.

Definitions 2.15 and 2.16 refers to a Σ_r-theory T_r such that T is T_r-compatible. Although this is not relevant for the proofs of the results in this paper, we notice that if such a theory T_r exists, then one can always take T_r to be the theory axiomatized by the universal Σ_r-sentences which are logical consequences of T.

3 The Satisfiability Problem

We completed our conceptual setting: we need however to restrict it considerably, in order to be able to provide positive results. This is partially done by means of the following further assumption, to be kept in mind for the whole paper.

[12] For instance, the theory of dense linear orders is locally finite but cannot be the theory of a single finite structure, because finite linear orders are not dense.

Assumption 3.1. We shall concentrate on *ground* satisfiability problems. For this reason, we assume *the underlying theory T of an LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ to have decidable constraint satisfiability problem.*

We will see that this assumption alone is not sufficient to guarantee the decidability of the ground satisfiability problem for LTL-theories (cf. Section 3.1). Fortunately, the problem becomes decidable (cf. Sections 3.2 and 3.3) when the underlying theory T of an LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ satisfies the same requirements for the correctness of the combination schema of Section 2.2.3.

3.1 Undecidability

We show that the decidability of the satisfiability problem for (totally flexible) LTL-theories implies the decidability of the constraint satisfiability problem for unions of (signature disjoint) theories in a first-order framework. This reduction proves undecidability, as shown in [5] (in fact, both recent and long standing literature [36, 44] impose further requirements, such as stable infiniteness, on the component theories to obtain positive decidability transfer results of the constraint satisfiability problem).

Theorem 3.2. *There exists a totally flexible LTL-theory $\mathcal{T}$ whose ground satisfiability problem is undecidable.*

There are two key observations underlying the proof of this undecidability result. First, we build a theory T whose constraint satisfiability problem consists of non-deterministically solving the constraint satisfiability problem among two signature-disjoint theories T_1, T_2. It is easy to see that the constraint satisfiability problem of T is decidable, if it is decidable for both T_1 and T_2. The second observation is that it is possible to write an LTL($\Sigma^{\underline{a}}$)-sentence whose satisfiability is equivalent to the satisfiability of a constraint in $T_1 \cup T_2$. In [5], it is shown that such a problem is undecidable for suitable T_1 and T_2.

Proof. We must define an LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ such that $\Sigma_r = \emptyset$, i.e. $\mathcal{T}$ is totally flexible, and the constraint satisfiability problem of T is decidable, according to Assumption 3.1.

To define a suitable T, the following two facts about combinations of theories are crucial.

(i) There exist theories T_1, T_2 whose constraint satisfiability problem is decidable, whose signatures Σ_1, Σ_2 are disjoint and such that the constraint satisfiability problem of $T_1 \cup T_2$ is undecidable (this is shown in [5]).

(ii) Let T be a Σ-theory whose constraint satisfiability problem is decidable and Σ' be a signature such that $\Sigma' \supseteq \Sigma$. If we consider T as a Σ'-theory, then the constraint satisfiability problems of T is still decidable (this is proved in, e.g., [21, 45]).

Consider now theories T_1, T_2 as in (i) above and let us define a new Σ-theory T as follows:

$$\Sigma := \Sigma_1 \cup \Sigma_2 \cup \{P\} \quad \text{and} \quad T := \{P \rightarrow \psi \mid \psi \in T_1\} \cup \{\neg P \rightarrow \psi \mid \psi \in T_2\},$$

where P is a fresh 0-ary predicate symbol (or, otherwise said, a fresh propositional letter). We claim that the constraint satisfiability problem for the Σ-theory T is decidable. In fact, given a $\Sigma_1 \cup \Sigma_2 \cup \{P\}$ constraint Γ, we first guess the truth value of P and add either P or $\neg P$ to Γ, accordingly. At this point, we are left with the problem of solving a constraint satisfiability problem of the $(\Sigma_1 \cup \Sigma_2 \cup \{P\})$-theory T_i for either $i = 1$ or $i = 2$. This is decidable by fact (ii) above: the constraint satisfiability problem of the Σ_i-theory T_i is decidable by assumption and the symbols from $\Sigma_j \cup \{P\}$ $(j \neq i)$ are free for T_i.

We now show that the ground satisfiability problem for $\mathcal{T}$ is undecidable by identifying a particular class of ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentences whose satisfiability cannot be decided. We assume that there are infinitely many system parameters (whereas the cardinality of the set of system variables is irrelevant). We claim that it is not possible to decide the $\mathcal{T}$-satisfiability of the following type of ground LTL($\Sigma^{\underline{c}}$)-sentences:

$$P \wedge \Gamma_1 \wedge X(\neg P \wedge \Gamma_2), \tag{2}$$

where Γ_i is a finite conjunction of $\Sigma_i^{\underline{c}}$-literals (for $i = 1, 2$) and the $\underline{c}$ are the free constants of the LTL-theory $\mathcal{T}$ (i.e. the rigid system parameters). In fact, if (2) is satisfiable (in the sense of Definition 2.13) then it is easy to build a model (in first-order semantics) for $T_1 \cup T_2$ satisfying $\Gamma_1 \cup \Gamma_2$, and also the converse holds. Thus the satisfiability of the sentences of the kind described in (2) is reduced to the satisfiability w.r.t. $T_1 \cup T_2$ of the arbitrary constraint $\Gamma_1 \cup \Gamma_2$: this is undecidable by fact (i) above (notice that the satisfiability of pure constraints, like $\Gamma_1 \cup \Gamma_2$ is equivalent to satisfiability of arbitrary $(\Sigma_1 \cup \Sigma_2)$-constraints, because every constraint is equi-satisfiable with an effectively built pure constraint, see **Step 1** from Section 2.2.3). □

3.2 Decidability and Locally Finite LTL-Theories

Let $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ be a given LTL-theory. The arguments underlying the proof of Theorem 3.2 suggest that the undecidability of the ground satisfiability problem for $\mathcal{T}$ arises precisely for the same reasons leading to the undecidability of combined constraint satisfiability problems in the first-order framework. The hope is that the same (or similar) requirements

yielding the decidability of the constraint satisfiability problem in unions of theories will also give the decidability of the ground satisfiability problem for $\mathcal{T}$. It turns out that this is indeed the case for both locally finite and noetherian theories (cf. Section 2.2.2).

Theorem 3.3. *The ground satisfiability problem for a locally finite compatible LTL-theory is decidable.*

Below, we give two constructive proofs of this Theorem. The former is based on an eager reduction to the satisfiability problem for propositional LTL. The latter consists in a lazy integration between a standard tableau algorithm for the satisfiability problem of propositional LTL and a decision procedure for the constraint satisfiability problem in the background (first-order) theory T.

3.2.1 Eager Reduction to Propositional LTL-Satisfiability

In the rest of this Subsection, let $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ be a locally finite compatible LTL-theory. We prove Theorem 3.3 by a reduction to satisfiability in propositional linear temporal logic (PLTL, from now on). The syntactic relationship between first-order and propositional LTL-formulae is given by the notion of abstraction.

Definition 3.4 (PLTL-Abstraction)**.** Given a signature $\Sigma^{\underline{a}}$ and a set of propositional letters $\mathcal{L}$ of the appropriate cardinality, let $[\![\cdot]\!]$ be a bijection from the set of ground $\Sigma^{\underline{a}}$-atoms into $\mathcal{L}$. By translating identically boolean and temporal connectives, the map is inductively extended to a bijective map (also called $[\![\cdot]\!]$) from the set of ground LTL($\Sigma^{\underline{a}}$)-sentences onto the set of propositional $\mathcal{L}$-formulae.

Given a ground LTL($\Sigma^{\underline{a}}$)-sentence φ, we call $[\![\varphi]\!]$ the *PLTL-abstraction* of φ. Given a set Θ of ground LTL($\Sigma^{\underline{a}}$)-sentences, $[\![\Theta]\!]$ denotes the set $\{[\![\varphi]\!] \mid \varphi \in \Theta\}$.

The following straightforward Lemma explains why PLTL-abstractions are relevant for satisfiability checking of LTL($\Sigma^{\underline{a}}$)-sentences.

Lemma 3.5. *Let $\mathcal{L}$ be a set of propositional letters, Σ be a signature, $\underline{a}$ be a set of free constants, and $[\![\cdot]\!]$ be a PLTL-abstraction function mapping ground LTL($\Sigma^{\underline{a}}$)-sentences into propositional $\mathcal{L}$-formulae. Suppose we are given a ground LTL($\Sigma^{\underline{a}}$)-sentence φ, a Kripke model V for $\mathcal{L}$ and an LTL($\Sigma^{\underline{a}}$)-structure $\mathcal{M} = \{\mathcal{M}_n\}_{n \in \mathbb{N}}$ such that for every $t \in \mathbb{N}$ and for every $\Sigma^{\underline{a}}$-ground atom ψ occurring in φ we have*

$$\mathcal{M}_t \models \psi \qquad \textit{iff} \qquad V_t([\![\psi]\!]) = 1.$$

Then we have also

$$\mathcal{M} \models_t \varphi \qquad \textit{iff} \qquad V \models_t [\![\varphi]\!],$$

for every $t \in \mathbb{N}$.

The key to define a reduction to the satisfiability problem in PLTL is guessing.

Definition 3.6 (Guessing)**.** Given a signature Σ and a finite set of Σ-atoms S, an *S-guessing* $\mathcal{G}$ is a boolean assignment to members of S (we view $\mathcal{G}$ as the set $\{\varphi \mid \varphi \in S$ and $\mathcal{G}(\varphi)$ is assigned to true$\} \cup \{\neg\varphi \mid \varphi \in S$ and $\mathcal{G}(\varphi)$ is assigned to false$\}$).

Indeed, guessing must take into account rigid constants: each guessing of atoms over flexible symbols must be "compatible" with the guessing of atoms over rigid symbols. Formally, this is ensured as follows.

By definition of locally finite compatible LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$, there must exist a theory T_r such that $T_r \subseteq T$ is effectively locally finite. So, given a finite subset $\underline{c}_0$ of $\underline{c}$, it is possible to compute a finite set S of ground $\Sigma_r^{\underline{c}_0}$-atoms which are representative modulo T-equivalence of all ground $\Sigma_r^{\underline{c}_0}$-atoms: for this choice of S, an S-guessing is called a *rigid $\underline{c}_0$-guessing*. Now, let $\tilde{S}$ be any finite set of $\Sigma^{\underline{a},\underline{c}}$-atoms and $\mathcal{G}$ be a rigid $\underline{c}_0$-guessing: an $\tilde{S}$-guessing $\tilde{\mathcal{G}}$ is *$\mathcal{G}$-compatible* iff $\mathcal{G} \cup \tilde{\mathcal{G}}$ is T-satisfiable. The set of $\mathcal{G}$-compatible $\tilde{S}$-guessing is denoted by $C(\tilde{S}, \mathcal{G})$.

Theorem 3.3 is an immediate consequence of the well-known fact that PLTL-satisfiability is decidable and the following Proposition.

Proposition 3.7. *Let $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ be a locally finite compatible LTL-theory. Let $\mathcal{L}$ be a set of propositional letters and $[\![\cdot]\!]$ be a PLTL-abstraction function mapping ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentences into propositional $\mathcal{L}$-formulae. A ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence φ is satisfiable in an LTL($\Sigma^{\underline{a},\underline{c}}$)-structure $\mathcal{M}$ appropriate for $\mathcal{T}$ iff there exists a rigid $\underline{c}_0$-guessing $\mathcal{G}$ such that the propositional formula*

$$[\![\varphi]\!] \wedge \Box \bigwedge_{\psi \in \mathcal{G}} [\![\psi]\!] \wedge \Box \Big(\bigvee_{\tilde{\mathcal{G}} \in C(At(\varphi), \mathcal{G})} \bigwedge_{\psi \in \tilde{\mathcal{G}}} [\![\psi]\!] \Big) \tag{3}$$

is satisfiable in a PLTL-Kripke model (here $\underline{c}_0$ is the subset of the set $\underline{c}$ of system parameters occurring in φ and $At(\varphi)$ is the set of $\Sigma^{\underline{a},\underline{c}}$-atoms occurring in φ).

Proof. The 'only if' is immediate from Lemma 3.5. The converse can be derived from Lemma A.7 from the Appendix. Suppose that the PLTL-formula (3) is satisfiable in a Kripke model $V = \{V_n\}_{n \in \mathbb{N}}$ for a certain rigid $\underline{c}_0$-guessing $\mathcal{G}$. This means that for every n there is $\tilde{\mathcal{G}}_n \in C(At(\varphi), \mathcal{G})$ such that $V \models_n \bigwedge_{\psi \in \mathcal{G}} [\![\psi]\!] \wedge \bigwedge_{\psi \in \tilde{\mathcal{G}}_n} [\![\psi]\!]$. Since $\tilde{\mathcal{G}}_n$ is $\mathcal{G}$-compatible, there is a $\Sigma^{\underline{a},\underline{c}_0}$-structure $\mathcal{M}_n$ which is a model of $T \cup \tilde{\mathcal{G}}_n \cup \mathcal{G}$; by Lemma A.7, the $\mathcal{M}_n$ can be $\Sigma^{\underline{a},\underline{c}_0}$-embedded into $\Sigma^{\underline{a},\underline{c}}$-structures $\mathcal{M}'_n$ such that $\mathcal{M}' := \{\mathcal{M}'_n\}_{n \in \mathbb{N}}$ is appropriate for $\mathcal{T}$.[13] The

[13] Lemma A.7 is used with $I := \mathbb{N}$, and $T_i := T$, but symbols from $\Sigma \setminus \Sigma_r$ are disjointly renamed when

$\mathcal{M}_n$ can be seen as $\Sigma^{\underline{a},\underline{c}}$-structures by interpreting rigid parameters $\underline{c} \setminus \underline{c}_0$ arbitrarily (but in the same way in all $\mathcal{M}_n$). Since truth of ground literals is preserved through embeddings, $\mathcal{M}'_n$ is again a model of $\tilde{\mathcal{G}}_n$ for every n. But then Lemma 3.5 ensures that $\mathcal{M}' \models \varphi$, given that $V \models [\![\varphi]\!]$. □

Example 3.8 ([38])**.** Let $\mathcal{T} = \langle \{>\}, T_{lo}, \{>\}, \underline{a}, \underline{c} \rangle$ be an LTL-theory, where T_{lo} is the theory of strict linear orders and $>$ is a binary predicate symbol. Since T_{lo} (i) is universal, (ii) admits as a model completion the theory of dense linear order without endpoints and (iii) is effectively locally finite, then $\mathcal{T}$ is a locally finite compatible LTL-theory; moreover, it is easy to check that the constraint satisfiability problem for T_{lo} is decidable. We are interested to check the satisfiability of the following LTL($\Sigma^{\underline{a}}$)-sentence:[14]

$$\varphi \quad :\equiv \quad a > b \wedge b > c \wedge (\Diamond a = c \vee \Diamond c > a)$$

Indeed, the solution to this satisfiability problem depends on how we classify the symbols a, b, and c. Notice that the set $At(\varphi)$ of atoms in φ is $\{a > b, b > c, a = c, c > a\}$. Now, let us consider two cases according to how a, b, c are considered as flexible or rigid.

1. $\underline{a} = \{b\}$ and $\underline{c} = \{a, c\}$. The set of representative $\Sigma^{\underline{c}}$-atoms is $\{a > c, a = c, c > a\}$. The rigid $\underline{c}$-guessings which are consistent w.r.t. T_{lo} are therefore the following:

$$\begin{aligned} \mathcal{G}_1 :=& \{a > c, a \neq c, c \not> a\}, \\ \mathcal{G}_2 :=& \{a \not> c, a = c, c \not> a\}, \\ \mathcal{G}_3 :=& \{a \not> c, a \neq c, c > a\}. \end{aligned}$$

   We omitted to consider the rigid $\underline{c}$-guessings which are not T_{lo}-satisfiable because every T_{lo}-unsatisfiable $\underline{c}$-guessing $\mathcal{G}$ leads to the inconsistency of the formula (3) since there is no $\mathcal{G}$-compatible $At(\varphi)$-guessing. Consider now the first two conjuncts of (3) for each $\mathcal{G}_i$:

   $\mathcal{G}_1$**:** from

$$\begin{aligned} [\![a > b]\!] \wedge [\![b > c]\!] \wedge (\Diamond [\![a = c]\!] \vee \Diamond [\![c > a]\!]) \wedge \\ \wedge \Box([\![a > c]\!] \wedge \neg [\![a = c]\!] \wedge \neg [\![c > a]\!]) \end{aligned}$$

building the signature Σ_i for the i-th copy of T (the same observation applies also to the flexible constants $\underline{a}$). In this way, a model of $\bigcup_i T_i$ is the same thing as a sequence of models $\{\mathcal{M}'_n\}_{n \in \mathbb{N}}$ of T whose Σ_r-reducts coincide.

[14] The formula is obtained by negating $a > b \wedge b > c \rightarrow \Box(a > c)$

we obtain

$$(\llbracket a > b \rrbracket \wedge \llbracket b > c \rrbracket \wedge \underline{\Diamond \llbracket a = c \rrbracket} \wedge \Box \llbracket a > c \rrbracket \wedge \underline{\Box \neg \llbracket a = c \rrbracket} \wedge \Box \neg \llbracket c > a \rrbracket) \vee$$
$$\vee (\llbracket a > b \rrbracket \wedge \llbracket b > c \rrbracket \wedge \underline{\Diamond \llbracket c > a \rrbracket} \wedge \Box \llbracket a > c \rrbracket \wedge \Box \neg \llbracket a = c \rrbracket \wedge \underline{\Box \neg \llbracket c > a \rrbracket})$$

Each disjunct is easily found PLTL-unsatisfiable because of the inconsistency between the underlined part of the formula.

$\mathcal{G}_2$**:** from

$$\llbracket a > b \rrbracket \wedge \llbracket b > c \rrbracket \wedge (\Diamond \llbracket a = c \rrbracket \vee \Diamond \llbracket c > a \rrbracket) \wedge$$
$$\wedge \Box (\neg \llbracket a > c \rrbracket \wedge \llbracket a = c \rrbracket \wedge \neg \llbracket c > a \rrbracket)$$

we obtain

$$(\llbracket a > b \rrbracket \wedge \llbracket b > c \rrbracket \wedge \Diamond \llbracket a = c \rrbracket \wedge \Box \neg \llbracket a > c \rrbracket \wedge \Box \llbracket a = c \rrbracket \wedge \Box \neg \llbracket c > a \rrbracket) \vee$$
$$\vee (\llbracket a > b \rrbracket \wedge \llbracket b > c \rrbracket \wedge \underline{\Diamond \llbracket c > a \rrbracket} \wedge \Box \neg \llbracket a > c \rrbracket \wedge \Box \llbracket a = c \rrbracket \wedge \underline{\Box \neg \llbracket c > a \rrbracket})$$

The second disjunct is easily found PLTL-unsatisfiable because of the inconsistency between the underlined part of the formula. We are left to check the PLTL-unsatisfiable of the following formula obtained by considering all $\mathcal{G}_2$-compatible guessings:

$$\underline{\llbracket a > b \rrbracket} \wedge \underline{\underline{\llbracket b > c \rrbracket}} \wedge \Diamond \llbracket a = c \rrbracket \wedge \Box \neg \llbracket a > c \rrbracket \wedge \Box \llbracket a = c \rrbracket \wedge \Box \neg \llbracket c > a \rrbracket \wedge$$
$$\wedge \Box \left(\begin{array}{cc} & (\underline{\neg \llbracket a > b \rrbracket} \wedge \llbracket b > c \rrbracket \wedge \llbracket a = c \rrbracket \wedge \neg \llbracket c > a \rrbracket) \quad \vee \\ \vee & (\llbracket a > b \rrbracket \wedge \underline{\underline{\neg \llbracket b > c \rrbracket}} \wedge \llbracket a = c \rrbracket \wedge \neg \llbracket c > a \rrbracket) \quad \vee \\ \vee & (\underline{\neg \llbracket a > b \rrbracket} \wedge \underline{\underline{\neg \llbracket b > c \rrbracket}} \wedge \llbracket a = c \rrbracket \wedge \neg \llbracket c > a \rrbracket) \end{array} \right)$$

which is easily found PLTL-inconsistent by observing the underlined literals.

$\mathcal{G}_3$**:** from

$$\llbracket a > b \rrbracket \wedge \llbracket b > c \rrbracket \wedge (\Diamond \llbracket a = c \rrbracket \vee \Diamond \llbracket c > a \rrbracket) \wedge$$
$$\wedge \Box (\neg \llbracket a > c \rrbracket \wedge \neg \llbracket a = c \rrbracket \wedge \llbracket c > a \rrbracket)$$

we obtain

$$(\llbracket a > b \rrbracket \wedge \llbracket b > c \rrbracket \wedge \underline{\Diamond \llbracket a = c \rrbracket} \wedge \Box \neg \llbracket a > c \rrbracket \wedge \underline{\Box \neg \llbracket a = c \rrbracket} \wedge \Box \llbracket c > a \rrbracket) \vee$$
$$\vee (\llbracket a > b \rrbracket \wedge \llbracket b > c \rrbracket \wedge \Diamond \llbracket c > a \rrbracket \wedge \Box \neg \llbracket a > c \rrbracket \wedge \Box \neg \llbracket a = c \rrbracket \wedge \Box \llbracket c > a \rrbracket)$$

The first disjunct is easily found PLTL-unsatisfiable because of the inconsistency between the underlined part of the formula. We are left to check the PLTL-unsatisfiable of the following formula obtained by considering all the $\mathcal{G}_3$-compatible guessings:

$$\underline{[\![a > b]\!]} \wedge \underline{\underline{[\![b > c]\!]}} \wedge \Diamond [\![c > a]\!] \wedge \Box \neg [\![a > c]\!] \wedge \Box \neg [\![a = c]\!] \wedge \Box [\![c > a]\!] \wedge$$
$$\wedge \Box \left(\begin{array}{lll} & \left(\neg \underline{[\![a > b]\!]} \wedge [\![b > c]\!] \wedge \neg [\![a = c]\!] \wedge [\![c > a]\!] \right) & \vee \\ \vee & \left([\![a > b]\!] \wedge \neg \underline{\underline{[\![b > c]\!]}} \wedge \neg [\![a = c]\!] \wedge [\![c > a]\!] \right) & \vee \\ \vee & \left(\neg \underline{[\![a > b]\!]} \wedge \neg \underline{\underline{[\![b > c]\!]}} \wedge \neg [\![a = c]\!] \wedge [\![c > a]\!] \right) & \end{array} \right)$$

which is easily found PLTL-inconsistent by observing the underlined literals.

Since there is no rigid guessing such that the formula (3) is PLTL-satisfiable, we are entitled to conclude that φ is unsatisfiable in any LTL($\Sigma^{\{b\},\{a,c\}}$)-structure appropriate for $\mathcal{T}$.

2. $\underline{a} = \{a, b, c\}$ and $\underline{c} = \emptyset$. Since there are no system parameters, all the $At(\varphi)$-guessings which are T_{lo}-satisfiable are trivially compatible with every rigid $\underline{c}$-guessing. It easy to check that the corresponding instance of (3) is PLTL-satisfiable. Hence, by Theorem 3.3, we conclude that φ is satisfiable in an LTL($\Sigma^{\{a,b,c\},\emptyset}$)-structure appropriate for $\mathcal{T}$.

Proposition 3.7 gives an algorithm to solve the ground satisfiability problem for $\mathcal{T}$, when $\mathcal{T}$ is a locally finite compatible LTL-theory. For the PLTL-satisfiability test, one may use any decision procedure for PLTL-satisfiability based on tableaux, automata, or temporal extensions of resolution. Such an algorithm can be regarded as an *eager reduction algorithm*, in the sense that it produces an instance of a PLTL-satisfiability problem. The drawback is that the resulting PLTL-satisfiability problem may be quite large. The main advantage is that both decision procedures for the constraint satisfiability problem of the underlying locally finite theory and decision procedures for PLTL can be used 'off-the-shelf'. In the next Subsection, we consider a procedure which is likely to scale up more smoothly at the price of a finer grain integration between the constraint reasoner in the background theory and the PLTL satisfiability solver.

3.2.2 A Lazy Tableau Procedure

We describe a *lazy* approach to solve the ground satisfiability problem for LTL-theories by extending the classic approach to temporal propositional satisfiability adopted in the Tableaux community. The key idea is to lift the definition of Hintikka sets to ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentences of the form (3). The soundness and completeness proof of the resulting algorithm

(cf. Corollary 3.12 below) is immediate from Proposition 3.7 and basic properties of tableaux for PLTL (see, e.g., [33], Section 5.5).

As before, let us fix a locally finite compatible LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$. A ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence is in *Negation Normal Form* (NNF) iff it is built up from LTL($\Sigma^{\underline{a},\underline{c}}$)-literals by using $\vee, \wedge, X, \Box, U$. It can be shown that every ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence is logically equivalent to a formula in NNF.[15]

Definition 3.9. Given a ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence φ in NNF, the *closure* of φ is the set $cl(\varphi)$ containing:

(i) all subformulae of φ and all negations of atoms occurring in φ;

(ii) a representative set (modulo T-equivalence) of $\Sigma_r^{\underline{c}_0}$-literals, where $\underline{c}_0$ is the finite set of system parameters occurring in φ;

(iii) formulae of the form $X(\psi U \chi)$, where $\psi U \chi$ is a subformula of φ;

(iv) formulae of the form $X\Box\psi$, where $\Box\psi$ is a subformula of φ.

Notice that $cl(\varphi)$ is finite and has cardinality $O(\max(n, k(\underline{c}_0)))$, if n is the length of φ and $k(\underline{c}_0)$ is the cardinality of a representative set of $\Sigma_r^{\underline{c}_0}$-literals.

Definition 3.10. Given a ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence φ in NNF, a *Hintikka set* for φ is a subset $H \subseteq cl(\varphi)$ such that:

(i) for every atom $\psi \in cl(\varphi)$, H contains either ψ or $\neg\psi$;

(ii) the set of literals from H is T-satisfiable;

(iii) if $\psi_1 \wedge \psi_2 \in H$, then ($\psi_1 \in H$ and $\psi_2 \in H$);

(iv) if $\psi_1 \vee \psi_2 \in H$, then ($\psi_1 \in H$ or $\psi_2 \in H$);

(v) if $\psi_1 U \psi_2 \in H$, then ($\psi_2 \in H$ or ($\psi_1 \in H$ and $X(\psi_1 U \psi_2) \in H$));

(vi) if $\Box\psi \in H$, then ($\psi \in H$ and $X\Box\psi \in H$).

We are now in the position to define a Hintikka graph, which will be used as the key data structure to define the tableau procedure.

Definition 3.11. The *Hintikka graph* $\mathcal{H}(\varphi)$ of φ is the directed graph having as nodes the Hintikka sets for φ and as edges the pairs $H \rightarrow H'$ such that

[15] For simplicity (and ignoring efficiency problems), we include $\Box$ but not the 'release operator' $\varphi R \psi := \neg(\neg\varphi U \neg\psi)$ (this operator can be defined in terms of $\Box$ and U as $\Box\psi \vee (\psi U(\varphi \wedge \psi))$).

(i) $H' \supseteq \{\psi \mid X\psi \in H\}$;

(ii) H and H' contain the same ground $\Sigma_r^{\underline{c}_0}$-literals.

A *strongly connected subgraph* (scs) of $\mathcal{H}(\varphi)$ is a set $\mathcal{C}$ of nodes of $\mathcal{H}(\varphi)$ such that for every $H, H' \in \mathcal{C}$ there is a (non-empty) $\mathcal{H}(\varphi)$-path from H to H' whose nodes all belong to $\mathcal{C}$.[16] An scs $\mathcal{C}$ is *fulfilling* [33] iff for every $\psi_1 U \psi_2 \in cl(\varphi)$ there is $H \in \mathcal{C}$ such that either $\psi_1 U \psi_2 \notin H$ or $\psi_2 \in H$. A node H in $\mathcal{H}(\varphi)$ is *initial* iff $\varphi \in H$.

Corollary 3.12. *A ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence φ in NNF is satisfiable in an LTL($\Sigma^{\underline{a},\underline{c}}$)-structure $\mathcal{M}$ appropriate for $\mathcal{T}$ iff there is an $\mathcal{H}(\varphi)$-path leading from an initial node into a fulfilling scs.*

An observation about the complexity of the lazy procedure is in order. When the set of representative $\Sigma_r^{\underline{c}_0}$-atoms has polynomial size, one can derive a PSPACE-decision procedure (provided that the T-constraint satisfiability problem is also PSPACE) from the above Corollary. The crucial point is to avoid the explicit construction of the Hintikka graph and explore it 'on-the-fly' by using well-known techniques of the PLTL literature (see, e.g., [41])

3.3 Decidability and Noetherian LTL-Theories

We now extend Theorem 3.3 to the noetherian compatible case. This is a more difficult task: since no apriori rigid guessing can be performed, we shall need positive constraint propagation along the model to be built.

Suppose α is a PLTL-formula in NNF; we define the closure $cl_P(\alpha)$ of α, the notion of a Hintikka set for α, and the Hintikka graph $\mathcal{H}_P(\alpha)$ for α, simply by dropping clause (ii) from Definitions 3.9, 3.10, 3.11 (we use the subscript 'P' to stress that these definitions refer to PLTL-formulae and not to ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentences, like in the previous subsection). We can similarly define initial nodes, scs's, and fulfilling scs's for $\mathcal{H}_P(\alpha)$.

Definition 3.13 (pc-model)**.** Given a PLTL-formula α in NNF, a pair $\langle \mathcal{P}, \mathcal{C} \rangle$ is a pc-model for α iff

(i) $\mathcal{P}$ is a path $H_0 \to H_1 \to \cdots \to H_k$ in $\mathcal{H}_P(\alpha)$ such that H_0 is an initial node of $\mathcal{H}_P(\alpha)$ and the nodes $H_1, \ldots, H_k$ are all distincts;

(ii) $\mathcal{C}$ is a fulfilling scs containing H_k.

Notice that, given a PLTL-formula α, there are only finitely many pc-models for α. Corollary 3.12 now reads as the following well-known result (cf., e.g., [33]):[17]

[16] In particular, for $H = H'$, we see that an scs cannot consist of a single not self-accessible node.

[17] Theorem 3.14 (like Corollary 3.12) can be refined by asking the scs of a pc-model to be maximal.

Theorem 3.14. *A PLTL-formula α in NNF is satisfiable iff there is a pc-model for α.*

3.3.1 The Procedure NSat

Let $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ be a noetherian compatible LTL-theory. The procedure NSAT takes in input a ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence φ and returns *"satisfiable"* if there is an LTL($\Sigma^{\underline{a},\underline{c}}$)-structure $\mathcal{M}$ appropriate for $\mathcal{T}$ such that $\mathcal{M} \models \varphi$, *"unsatisfiable"* otherwise.

The procedure NSAT relies on a procedure DP-LTL which, given a ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence φ, is able to enumerate all pc-models $\langle \mathcal{P}, \mathcal{C} \rangle$ for $[\![\varphi]\!]$. Moreover, the decision procedure DP-T is a $DPLL(\mathcal{T})$-based decision procedure for the constraint satisfiability problem for the theory T (i.e., it is able to cope with T-satisfiability of sets of ground $\Sigma^{\underline{a},\underline{c}}$-clauses instead of only sets of ground $\Sigma^{\underline{a},\underline{c}}$-literals). Finally, $Res_T^{\underline{c}}$ is the T-residue enumerator for T_r w.r.t. $\underline{c}$. In the outer loop of the NSAT procedure, pc-models for $[\![\varphi]\!]$ are enumerated; let

Algorithm 1 The satisfiability procedure for the noetherian compatible case

Require: φ ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence

```
1:  procedure NSAT(φ)
2:      for all ⟨P, C⟩ ∈ DP-LTL([[ φ ]]) do
3:          B' ← ∅
4:          repeat
5:              B ← B'
6:              for all H_i ∈ ⟨P, C⟩ do
7:                  B_i ← Res_T^c(Γ_i ∪ B)        ▷ Γ_i := {ℓ | ℓ is a literal and [[ ℓ ]] ∈ H_i}
8:              end for
9:              B' ← ⋃_i B_i
10:         until DP-T(B ∧ ¬B') = "unsatisfiable"
11:         if DP-T(B) = "satisfiable" then
12:             return "satisfiable"
13:         end if
14:     end for
15:     return "unsatisfiable"
16: end procedure
```

$\langle \mathcal{P}, \mathcal{C} \rangle$ be the current one and let $H_1, \ldots, H_n$ be the Hintikka sets occurring in either $\mathcal{P}$ or $\mathcal{C}$. For each $i = 1, \ldots, n$, the procedure sets

$$\Gamma_i := \{\ell \mid \ell \text{ is a literal and } [\![\ell]\!] \in H_i\}$$

and begins the inner loop. In the inner loop, the local variable $\mathcal{B}$ is initialized to the empty set

and updated as follows: for $i = 1, \ldots, n$, the sets $\mathcal{B}_i$ which are the T_r-bases for $\Gamma_i \cup \mathcal{B}$ w.r.t. $\underline{c}$ are computed and the new value $\mathcal{B}'$ of $\mathcal{B}$ is $\bigcup_i \mathcal{B}_i$. If $\mathcal{B}'$ is T-consistent and logically equivalent to $\mathcal{B}$ modulo T, the procedure stops and returns *"satisfiable"*; if $\mathcal{B}'$ is not T-consistent, the procedure exit the inner loop and tries with another pc-model; finally, if $\mathcal{B}'$ is T-consistent but not logically equivalent to $\mathcal{B}$ modulo T, the inner loop is continued.

3.3.2 Correctness of NSat

Theorem 3.15 (Termination)**.** *The procedure* NSAT *always terminates.*

Proof. Since DP-T is a decision procedure, and DP-LTL enumerates a finite number of pc-models for $[\![\varphi]\!]$, they terminate. So, it remains to prove that the inner loop of lines 4-10 of Algorithm 1 terminates; to this aim we recall the fact (proved in Lemma A.11) that every infinite ascending chain of sets of positive ground $\Sigma^{\underline{c}}_r$-clauses is eventually constant for logical consequence w.r.t. a noetherian theory T_r. The test on line 10 eventually have to succeed by the following reason: if we let $\mathcal{B}^0, \mathcal{B}^1, \mathcal{B}^2, \ldots$ be the values of the local variable $\mathcal{B}$ after each execution of the loop, we have that $T_r \cup \mathcal{B}^{i+1} \models \mathcal{B}^i$, for each i, by Definition 2.4(ii). Thus, if we let $\mathcal{D}_i := \bigcup_{j \leq i} \mathcal{B}_j$, then the succession

$$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \ldots$$

is increasing and hence eventually constant modulo $T_r \subseteq T$, which means that also the above mentioned test eventually succeeds. □

Theorem 3.16 (Soundness)**.** *Let $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ be a noetherian compatible LTL-theory and φ be a ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence. If* NSAT*($\varphi$) returns "satisfiable", then there is an LTL($\Sigma^{\underline{a},\underline{c}}$)-structure $\mathcal{M}$ appropriate for $\mathcal{T}$ such that $\mathcal{M} \models \varphi$.*

Proof. If NSAT(φ) returns *"satisfiable"*, then for some pc-model $\langle \mathcal{P}, \mathcal{C} \rangle$ for $[\![\varphi]\!]$ (containing the n Hintikka sets $H_1, \ldots, H_n$), NSAT(φ) will produce the list of sets of positive ground $\Sigma^{\underline{c}}_r$-clauses

$$\mathcal{B}^0_1, \ldots, \mathcal{B}^0_n, \mathcal{B}^1_1, \ldots, \mathcal{B}^1_n, \ldots, \mathcal{B}^h_1, \ldots, \mathcal{B}^h_n,$$

such that:

- $\mathcal{B}^0, \ldots, \mathcal{B}^h, \mathcal{B}^{h+1}$ are the values of the local variable $\mathcal{B}$ in the iterations of the inner loop (we have $\mathcal{B}^0 = \emptyset, \mathcal{B}^1 = \bigcup_i \mathcal{B}^0_i, \ldots, \mathcal{B}^{h+1} = \bigcup_i \mathcal{B}^h_i$);
- for $j = 0, \ldots, h$ and for $i = 1, \ldots, n$, the set $\mathcal{B}^j_i$ is a T_r-basis for $\Gamma_i \cup \mathcal{B}^j$ w.r.t. $\underline{c}$ (here $\Gamma_i := \{\ell \mid \ell \text{ is a literal and } [\![\ell]\!] \in H_i\}$);
- $\mathcal{B}^{h+1}$ is T-consistent and logically equivalent to $\mathcal{B}^h$ modulo T.

Let $\mathcal{B} := \{C \mid T \cup \mathcal{B}^h \models C$ and C is a positive ground $\Sigma^{\underline{c}}_r$-clause$\}$; notice that $\mathcal{B}$ does not contain the empty clause, moreover we claim that for every positive ground $\Sigma^{\underline{c}}_r$-clause C and for each $i \in \{0, \ldots, n\}$, we have

$$T \cup \Gamma_i \cup \mathcal{B} \models C \quad \Rightarrow \quad C \in \mathcal{B}. \tag{4}$$

In fact, if $T \cup \Gamma_i \cup \mathcal{B} \models C$, then $T \cup \Gamma_i \cup \mathcal{B}^h \models C$ and so, by Definition 2.4(ii) $T_r \cup \mathcal{B}^h_i \models C$; but then $T_r \cup \mathcal{B}^{h+1} \models C$, meaning that $T \cup \mathcal{B}^h \models C$ (because $\mathcal{B}^{h+1}$ is logically equivalent to $\mathcal{B}^h$) and finally $C \in \mathcal{B}$ by the definition of the latter.

Recall that we are considering the pc-model $\langle \mathcal{P}, \mathcal{C} \rangle$; let us now choose a path $H_0 \to \cdots \to H_k \to \cdots \to H_{k+s}$ in $\mathcal{H}([\![\, \varphi \,]\!])$ such that $\{H_0 \to \cdots \to H_k\} = \mathcal{P}$ and $H_k \to \cdots \to H_{k+s}$ is a path in the scs $\mathcal{C}$ covering, possibly with repetitions, all the elements from it. We extend this finite path from H_0 to H_{k+s} into an infinite path

$$H_0 \to \cdots \to H_k \to \cdots \to H_{k+s} \to \cdots \to H_n \to \cdots$$

within $\mathcal{H}([\![\, \varphi \,]\!])$ by cyclically repeating $H_k, \ldots, H_{k+s}$ in the tail (that is, we take, for $i > k+s$, the Hintikka set H_i to be H_{k+p}, where p is the reminder of the integer division between $i-k$ and $s+1$).

By (4) and by Lemma A.8,[18] we obtain an infinite sequence $\mathcal{M}_0, \ldots, \mathcal{M}_i, \ldots$ of $\Sigma^{\underline{a},\underline{c}}$-structures such that (i) they all have the same support M and $\mathcal{M}_{i|\Sigma^{\underline{c}}_r} = \mathcal{M}_{j|\Sigma^{\underline{c}}_r}$ $(i, j \in \mathbb{N})$; (ii) $\mathcal{M}_i \models T \cup \Gamma_i$. These $\mathcal{M}_i$ consequently form an LTL$(\Sigma^{\underline{a},\underline{c}})$-structure $\mathcal{M} := \{\mathcal{M}_i\}_{i \in \mathbb{N}}$.

We show that $\mathcal{M} \models \varphi$, i.e. we prove by induction on the complexity of ψ (where $\psi \in cl(\varphi)$) that for every i it holds that:

$$[\![\, \psi \,]\!] \in H_i \quad \Rightarrow \quad \mathcal{M} \models_i \psi \tag{5}$$

In particular, we get $\mathcal{M} \models_0 \varphi$, because $[\![\, \varphi \,]\!] \in H_0$ (H_0 initial in $\mathcal{H}_P([\![\, \varphi \,]\!])$).

We need the following 'PLTL-like' argument (which is reported for the sake of completeness). The condition (5) is obvious if ψ is a literal or if it is of the kind $\psi_1 \wedge \psi_2$, $\psi_1 \vee \psi_2$ (see (ii) above and the definition of a Hintikka set for PLTL). If ψ is of the kind $X\psi_1$, then $[\![\, X\psi_1 \,]\!] \in H_i$ implies that $[\![\, \psi_1 \,]\!] \in H_{i+1}$, so it follows that $\mathcal{M} \models_{i+1} \psi_1$ by induction hypothesis, and thus $\mathcal{M} \models_i X\psi_1$ obtains. If ψ is of the kind $\Box\psi_1$, then $[\![\, \Box\psi_1 \,]\!] \in H_i$ implies $[\![\, \psi_1 \,]\!] \in H_j$ for each $j \geq i$, so it follows that $\mathcal{M} \models_j \psi_1$ for each $j \geq i$ by induction hypothesis, and thus $\mathcal{M} \models_i \Box\psi_1$.

Suppose now ψ is of the kind $\psi_1 U \psi_2$. Let us consider the following two cases:

- If $i < k$ there are two subcases to consider: (i) $[\![\, \psi_1 U \psi_2 \,]\!] \in H_k$ and $[\![\, \psi_1 \,]\!] \in H_j$ for every $i \leq j < k$; (ii) there exists $l < k$ such that $[\![\, \psi_2 \,]\!] \in H_l$ and $[\![\, \psi_1 \,]\!] \in H_j$ for every $i \leq j < l$.

[18] See the remark in footnote 13.

For the case (i) we can conclude that $\mathcal{M} \models_i \psi_1 U \psi_2$ by induction hypothesis and by the fact that $\mathcal{M} \models_k \psi_1 U \psi_2$ (see the case $i \geq k$ below), whereas for (ii) we can conclude by induction hypothesis that $\mathcal{M} \models_i \psi_1 U \psi_2$;

– If $i \geq k$, since $[\![\psi_1 U \psi_2]\!] \in H_i$ and since the scs $\mathcal{C}$ is fulfilling, there exists $H \in \mathcal{C}$ such that $[\![\psi_2]\!] \in H$.[19] Such an H occurs in the infinite list $H_i, H_{i+1}, \ldots$, because this list includes all the nodes from $\mathcal{C}$. Thus there exists the minimum $j \geq i$ such that $[\![\psi_2]\!] \in H_j$; for this j, the definition of a Hintikka set and of an edge in the Hintikka graph gives $[\![\psi_1]\!] \in H_l$ for every $i \leq l < j$, thus by induction hypothesis $\mathcal{M} \models_i \psi_1 U \psi_2$ obtains.

□

Theorem 3.17 (Completeness)**.** *Let $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ be a noetherian compatible LTL-theory and φ be a ground LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence. If there is an LTL($\Sigma^{\underline{a},\underline{c}}$)-structure $\mathcal{M}$ appropriate for $\mathcal{T}$ such that $\mathcal{M} \models \varphi$, then* NSAT*($\varphi$) returns "satisfiable".*

Proof. Let $\mathcal{M} = \{\mathcal{M}_n = (M, \mathcal{I}_n)\}_{n \in \mathbb{N}}$ be an LTL($\Sigma^{\underline{a},\underline{c}}$)-structure appropriate for $\mathcal{T}$ such that $\mathcal{M} \models \varphi$. Let us consider the sequence of the Hintikka sets $H_0, H_1, \ldots$ where

$$H_i = \{[\![\psi]\!] \in cl_P([\![\varphi]\!]) \mid \mathcal{M} \models_i \psi\}.$$

We obtain an infinite path $H_0 \to H_1 \to H_2 \to \cdots$ in $\mathcal{H}_P([\![\varphi]\!])$. Let H_k be the first Hintikka set which occurs infinitely many times in such a path. The set $\mathcal{C} := \{H_j \mid j \geq k\}$ is an scs of $\mathcal{H}_P([\![\varphi]\!])$; we prove that $\mathcal{C}$ is also fulfilling. To this aim, we take $[\![\psi_1 U \psi_2]\!] \in cl_P([\![\varphi]\!])$ and show that either $[\![\psi_1 U \psi_2]\!] \notin H_k$ or there is $j \geq k$ such that $[\![\psi_2]\!] \in H_j$: in fact this is obvious, because it means that either $\mathcal{M} \not\models_k \psi_1 U \psi_2$ or there is $j \geq k$ such that $\mathcal{M}_j \models \psi_2$.

We now take $\mathcal{P}$ to be a path from H_0 to H_k, whose nodes are pairwise distinct and included among $\{H_0, \ldots, H_k\}$. Thus the pair $\langle \mathcal{P}, \mathcal{C} \rangle$ is a pc-model for $[\![\varphi]\!]$ satisfying the following property:

(i) for every $H \in \mathcal{P} \cup \mathcal{C}$ there is i such that $H = \{[\![\psi]\!] \in cl_P([\![\varphi]\!]) \mid \mathcal{M} \models_i \psi\}$.

When examining the pc-model $\langle \mathcal{P}, \mathcal{C} \rangle$, the procedure DP-LTL produces a list of sets of positive ground $\Sigma_r^{\underline{c}}$-clauses

$$\mathcal{B}_1^0, \ldots, \mathcal{B}_n^0, \mathcal{B}_1^1, \ldots, \mathcal{B}_n^1, \ldots, \mathcal{B}_1^h, \ldots, \mathcal{B}_n^h,$$

such that:

[19] This is by the definition of a Hintikka set and of an edge in the graph $\mathcal{H}_P([\![\varphi]\!])$: notice that $[\![\psi_1 U \psi_2]\!]$ is inherited by all the nodes of a path within $\mathcal{H}_P([\![\varphi]\!])$ starting with H_i, unless the path meets a node to which $[\![\psi_2]\!]$ belongs. Now a path covering the whole $\mathcal{C}$ must meets such a node, because $\mathcal{C}$ is fulfilling.

– $\mathcal{B}^0, \ldots, \mathcal{B}^h, \mathcal{B}^{h+1}$ are the values of the local variable $\mathcal{B}$ in the iterations of the inner loop (we have $\mathcal{B}^0 = \emptyset, \mathcal{B}^1 = \bigcup_i \mathcal{B}_i^0, \ldots, \mathcal{B}^{h+1} = \bigcup_i \mathcal{B}_i^h$);

– for $j = 0, \ldots, h$ and for $i = 1, \ldots, n$, the set $\mathcal{B}_i^j$ is a T_r-basis for $\Gamma_i \cup \mathcal{B}^j$ w.r.t. $\underline{c}$, (here $\Gamma_i := \{\ell \mid \ell \text{ is a literal and } [\![\, \ell \,]\!] \in H_i\}$);

– $\mathcal{B}^{h+1}$ is logically equivalent to $\mathcal{B}^h$ modulo T.

We need to show that $\mathcal{B}^h$ is T-consistent. To this aim it is sufficient to observe (by induction on $j \leq h$) that the a $\Sigma_r^{\underline{c}}$-clause belonging to $\mathcal{B}^j$ is true in $\mathcal{M}_0$ (in fact in all the $\mathcal{M}_n$, because the symbols of $\Sigma_r^{\underline{c}}$ are rigidly interpreted): this is obvious for $j = 0$ and for $j > 0$ it is a direct consequence of (i) above, induction hypothesis and Definition 2.4(i). □

As an immediate corollary we obtain:

Theorem 3.18. *The ground satisfiability problem for a noetherian compatible LTL-theory is decidable.*

4 The Model-Checking Problem

4.1 LTL-System Specifications and the Model-Checking Problem

In order to introduce model-checking problems, we need some preliminary notions.

Definition 4.1. Given two signatures Σ_r and Σ such that $\Sigma_r \subseteq \Sigma$, we define the *one-step signature* as follows:

$$\Sigma \underset{\Sigma_r}{\oplus} \Sigma \quad := \quad ((\Sigma \setminus \Sigma_r) \uplus (\Sigma \setminus \Sigma_r)) \cup \Sigma_r,$$

where $\uplus$ denotes disjoint union.

In order to build the one-step signature $\Sigma \oplus_{\Sigma_r} \Sigma$, we first consider two copies of the symbols in $\Sigma \setminus \Sigma_r$; the two copies of $r \in \Sigma \setminus \Sigma_r$ are denoted by r^0 and r^1, respectively. Notice that the symbols in Σ_r are not renamed. Also, arities in the one-step signature $\Sigma \oplus_{\Sigma_r} \Sigma$ are defined in the obvious way: the arities of the symbols in Σ_r are unchanged and if n is the arity of $r \in \Sigma \setminus \Sigma_r$, then n is the arity of both r^0 and r^1. The one-step signature $\Sigma \oplus_{\Sigma_r} \Sigma$ will be also written as $\bigoplus_{\Sigma_r}^2 \Sigma$; similarly, we can define the n-step signature $\bigoplus_{\Sigma_r}^{n+1} \Sigma$ for $n > 1$ (our notation for the copies of $\Sigma \setminus \Sigma_r$-symbols extends in the obvious way, that is we denote by $r^0, r^1, \ldots, r^n$ the $n + 1$ copies of r).

Given an LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$, the one-step signature $\Sigma^{\underline{a},\underline{c}} \oplus_{\Sigma_r^{\underline{c}}} \Sigma^{\underline{a},\underline{c}}$ is appropriate to express constraints about the dynamic behavior of a system during one time unit.

In fact, the transition relation of a system must be able to simultaneously refer to the structures representing the state of the system in two consecutive time instants. In this respect, non-rigid symbols are duplicated whereas rigid symbols are left unchanged.

We now define the concepts of one-step structure and one-step theory, which are the semantic counterparts of the one-step signature introduced above (cf. Definition 4.1).

Definition 4.2. Given two signatures Σ_r and Σ such that $\Sigma_r \subseteq \Sigma$, two Σ-structures $\mathcal{M}_0 = \langle M, \mathcal{I}_0 \rangle$ and $\mathcal{M}_1 = \langle M, \mathcal{I}_1 \rangle$ whose Σ_r-reducts are the same,[20] the *one-step* $(\Sigma \oplus_{\Sigma_r} \Sigma)$*-structure*

$$\mathcal{M}_0 \underset{\Sigma_r}{\oplus} \mathcal{M}_1 = \langle M, \mathcal{I}_0 \underset{\Sigma_r}{\oplus} \mathcal{I}_1 \rangle$$

is defined as follows:

- for each function or predicate symbol $s \in \Sigma \setminus \Sigma_r$, we have $(\mathcal{I}_0 \oplus_{\Sigma_r} \mathcal{I}_1)(s^0) := \mathcal{I}_0(s)$ and $(\mathcal{I}_0 \oplus_{\Sigma_r} \mathcal{I}_1)(s^1) := \mathcal{I}_1(s)$;
- for each function or predicate symbol $r \in \Sigma_r$, we have $(\mathcal{I}_0 \oplus_{\Sigma_r} \mathcal{I}_1)(r) := \mathcal{I}_0(r)$.

If φ is a Σ-formula, the $(\Sigma \oplus_{\Sigma_r} \Sigma)$-formulae φ^0, φ^1 are obtained from φ by replacing each symbol $r \in \Sigma \setminus \Sigma_r$ by r^0 and r^1, respectively. The one-step theory is nothing but a combination of a theory T with a partially renamed copy of itself.

Definition 4.3. Given two signatures Σ_r and Σ such that $\Sigma_r \subseteq \Sigma$, the theory $T \oplus_{\Sigma_r} T$ is defined by $\{\varphi^0 \wedge \varphi^1 \mid \varphi \in T\}$.

We will write $\bigoplus^2_{\Sigma_r} T$ instead of $T \oplus_{\Sigma_r} T$; the n-step theories $\bigoplus^{n+1}_{\Sigma_r} T$ (for $n > 1$) are similarly defined.

Let now $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ be an LTL-theory whose parameters and whose system variables are *finite*. A *transition relation* for the LTL-theory $\mathcal{T}$ is a $(\Sigma^{\underline{a},\underline{c}} \oplus_{\Sigma^{\underline{c}}_r} \Sigma^{\underline{a},\underline{c}})$-sentence δ: we usually write such formula as $\delta(\underline{a}^0, \underline{a}^1)$ to emphasize that it contains the two copies of the system variables $\underline{a}$ (on the other hand, the system parameters $\underline{c}$ that are not duplicated will never be displayed). Examples of transition relations are the *tautological* transition $\delta_\top := \top$ and the *idle* transition:

$$\delta_I \quad := \quad \bigwedge_a (a^0 = a^1) \wedge \bigwedge_P \forall \underline{x} (P^0(\underline{x}) \leftrightarrow P^1(\underline{x})) \wedge \bigwedge_f \forall \underline{x} (f^0(\underline{x}) = f^1(\underline{x})),$$

where a ranges over free constants in $\underline{a}$, P over predicate symbols in $\Sigma \setminus \Sigma_r$, and f over function symbols in $\Sigma \setminus \Sigma_r$.

An *initial state description* for an LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ - with finitely many system variables and parameters - is simply a $\Sigma^{\underline{a},\underline{c}}$-sentence $\iota(\underline{a})$ (as it was the case for transition relations, the system parameters $\underline{c}$ will not be displayed also for state descriptions).

[20] Recall from Section 2 that this means that $\mathcal{I}_0(s) = \mathcal{I}_1(s)$ for all $s \in \Sigma_r$.

Definition 4.4 (LTL-System Specification and Model-Checking)**.** An *LTL-system specification* is an LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ (having finitely many system variables and system parameters) endowed with a transition relation $\delta(\underline{a}^0, \underline{a}^1)$ and with an initial state description $\iota(\underline{a})$. An LTL($\Sigma^{\underline{a},\underline{c}}$)-structure $\mathcal{M} = \{\mathcal{M}_n = (M, \mathcal{I}_n)\}_{n \in \mathbb{N}}$ is *a run* for such an LTL-system specification iff it is appropriate for $\mathcal{T}$ and moreover it obeys the transition δ and the initial state description ι, meaning that:

(i) $\mathcal{M}_n \oplus_{\Sigma^{\underline{c}}_r} \mathcal{M}_{n+1} \models \delta(\underline{a}^0, \underline{a}^1)$, for every $n \geq 0$;

(ii) $\mathcal{M}_0 \models \iota(\underline{a})$.

The *model-checking problem* for the system specification $(\mathcal{T}, \delta, \iota)$ is the following: given an LTL($\Sigma^{\underline{a},\underline{c}}$)-sentence φ, decide whether there is a run for $(\mathcal{T}, \delta, \iota)$ such that $\mathcal{M} \models \varphi$.[21] The *ground model-checking problem for* $(\mathcal{T}, \delta, \iota)$ is similarly introduced, but φ is assumed to be ground.

Roughly speaking, the satisfiability problem for LTL-theories (cf. Definition 2.13) is equivalent to the model-checking problem for LTL-system specifications endowed with tautological transition and tautological initial state description (there is a little difference, however, due to the fact that the satisfiability problem is relative to LTL-theories having possibly infinitely many system parameters and variables, whereas LTL-system specifications must have finitely many system variables and parameters).

An important subclass of model-checking problems is the following: the (syntactic) *safety model-checking problem* is the model-checking problem for formulae of the form

$$\Diamond \upsilon,$$

where υ is a $\Sigma^{\underline{a},\underline{c}}$-sentence. Since υ is meant to describe the set of *unsafe* states, we say that the system specification $(\mathcal{T}, \delta, \iota)$ is *safe for* υ iff the model-checking problem for $\Diamond\upsilon$ has a negative solution. This implies that $\Box\neg\upsilon$ is true for all runs of $(\mathcal{T}, \delta, \iota)$.

4.1.1 The Seriality Property

In the literature about model-checking (especially, for finite-state systems), it is usually assumed the seriality of the transition relation, i.e. that every state of the system must have

[21] Notice that usually the model-checking problem is taken to be the complement of our model-checking problem, i.e. it is taken to be the problem of deciding whether a given sentence is true in all runs. As far as we are concerned with decidability/undecidability issues, the difference is immaterial (for complexity questions, one must take the complementary classes). Our choice is motivated by the need of having a homogeneous terminology with satisfiability problems.

at least one successor state (see, e.g., [11] for more details). Unfortunately, it is difficult to find an effective formulation of such a requirement in our framework because the states of the system $(\mathcal{T}, \delta, \iota)$ are the models of the (first-order) theory underlying $\mathcal{T}$. Below, we give a non-effective formulation for seriality in our framework. Fortunately, as we shall see, there exist simple and effective methods to ensure it.

Definition 4.5. A system specification $(\mathcal{T}, \delta, \iota)$, based on the LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$, is said to be *serial* iff for every $\Sigma^{\underline{a},\underline{c}}$-structure $\mathcal{M}_0 = (M, \mathcal{I}_0)$ which is a model of T, there is another $\Sigma^{\underline{a},\underline{c}}$-structure $\mathcal{M}_1 = (M, \mathcal{I}_1)$ (still a model of T) such that $(\mathcal{M}_1)_{|\Sigma_r^{\underline{c}}} = (\mathcal{M}_2)_{|\Sigma_r^{\underline{c}}}$ and $\mathcal{M}_0 \oplus_{\Sigma_r^{\underline{c}}} \mathcal{M}_1 \models \delta(\underline{a}^0, \underline{a}^1)$.

In order to be able to ensure the above requirement for concrete situations, the following observations are useful:

(i) if the transition relation δ consists of the conjunction of (possibly guarded) assignments of the form $P \wedge a^1 = t^0(\underline{a}^0)$ where P is the condition under which the assignment is executed, then δ is serial (this is the case, for instance, of the water level controller example discussed in Section 4.4);

(ii) δ is serial when it is implied by the idle transition, i.e. in case $T \oplus_{\Sigma_r} T \models \delta_I \rightarrow \delta$ (this is equivalent to $T \models \delta^\sharp$, where $\delta^\sharp$ is obtained from δ by replacing the copies r^1, r^2 of every flexible symbol by r);[22]

(iii) every transition δ can be 'adjusted' in order to make it serial; to this end, it is sufficient to add a fresh 0-ary relational symbol E (standing for 'error') to Σ and replace δ by

$$\delta_E \quad := \quad (\neg E^0 \wedge \delta \wedge \neg E^1) \vee (\neg E^0 \wedge E^1) \vee (E^0 \wedge E^1).$$

4.1.2 Some Classes of LTL-Systems and further Assumptions

In Section 2.4.2, we have introduced three different classes of LTL-theories of increasing expressiveness so to study the satisfiability problem for LTL-theories. Here, we introduce the corresponding classes of LTL-systems so to study the decidability of the safety model-checking problem.

Definition 4.6. An LTL-system specification based on an LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ is said to be *finite state* iff $\mathcal{T}$ is totally rigid and T is an enumerated datatype theory.

Finite state system specifications are investigated by traditional symbolic model-checking literature [11] and are efficiently handled by state-of-the-art tools like `NuSMV` [10].

[22] If the constraint satisfiability problem of T is decidable and if δ is ground (as it is the case for some of the examples considered in this paper), the condition $T \models \delta^\sharp$ can be effectively checked.

Definition 4.7. An LTL-system specification based on an LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ is said to be *locally finite compatible* iff there is a Σ_r-universal and effectively locally finite theory T_r such that T is T_r-compatible.

As for compatible theories, from our discussion in Section 2.2.1, it follows that an LTL-system based on totally flexible LTL-theory is locally finite compatible in case its underlying theory is stably infinite.

Definition 4.8. An LTL-system specification based on an LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ is said to be *noetherian compatible* iff there is a Σ_r-universal theory T_r such that T is an effectively noetherian and T_r-compatible extension of T_r.

Since we are interested in positive results for safety model-checking problems, we need to make some restrictions; we first of all assume the following (which completes the Assumption from Section 3.1):

Assumption 4.9. For any LTL-system specification $(\mathcal{T}, \delta, \iota)$ (considered in the rest of this paper) we assume that: (i) the underlying theory of $\mathcal{T}$ has decidable constraint satisfiability problem; (ii) the transition relation δ and the initial state description ι are ground sentences; (iii) $(\mathcal{T}, \delta, \iota)$ is serial.

4.2 Undecidability and Noetherian LTL-Theories

Even under the above assumption, the ground model-checking problem for an LTL-system specification based on a totally rigid LTL-theory is undecidable: this is a folklore result that can be obtained through a simple reduction to the (undecidable) reachability problem of Minsky machines [35, 17]. We give details below for the sake of completeness.

A *two registers Minsky machine* is a finite set $\mathbf{P}$ of instructions (also called a program) for manipulating configurations seen as triples (s, m, n) of natural numbers, where s represents the machine state and m, n the contents of the two registers. There are four possible kinds of instructions, inducing transformations on the configurations as explained in Table 1. A $\mathbf{P}$-transformation is a transformation induced by an instruction of $\mathbf{P}$ on a certain configuration. For a Minsky machine $\mathbf{P}$, we write $(s, m, n) \rightarrow_{\mathbf{P}}^{\star} (s', m', n')$ to say that it is possible to reach configuration (s', m', n') from (s, m, n) by applying finitely many $\mathbf{P}$-transformations. Given a Minsky machine $\mathbf{P}$ and an initial configuration (s_0, m_0, n_0), the problem of checking whether a configuration (s', m', n') is reachable from (s_0, m_0, n_0) (i.e., if $(s_0, m_0, n_0) \rightarrow_{\mathbf{P}}^{\star} (s', m', n')$ holds or not) is called *the (second) reachability (configuration) problem*. It is well-known [8] that there exists a (two-register) Minsky machine $\mathbf{P}$ and a configuration (s_0, m_0, n_0) such that the second reachability configuration problem is undecidable.

N.	Instruction	Transformation
I	$s \to (t, 1, 0)$	$(s, m, n) \to (t, m+1, n)$
II	$s \to (t, 0, 1)$	$(s, m, n) \to (t, m, n+1)$
III	$s \to (t, -1, 0)[t']$	`if` $m \neq 0$ `then` $(s, m, n) \to (t, m-1, n)$ `else` $(s, m, n) \to (t', m, n)$
IV	$s \to (t, 0, -1)[t']$	`if` $n \neq 0$ `then` $(s, m, n) \to (t, m, n-1)$ `else` $(s, m, n) \to (t', m, n)$

Table 1: Instructions and related transformations for (two-registers) Minsky Machines

Theorem 4.10. *There exists a totally rigid and noetherian compatible LTL-system specification $(\mathcal{T}, \delta, \iota)$, whose ground safety model-checking problem is undecidable.*

Proof. The proof consists of two steps. First, we need to define a totally rigid LTL-theory $\mathcal{T}$ which is expressive enough to encode unbounded counters and which satisfies our Assumption (i) above. Second, we must define the encoding of a Minsky machine into an LTL-system based on $\mathcal{T}$ so that the second reachability problem of such machine can be represented as a safety model-checking problem. This immediately gives the undecidability of the latter, as desired.

Let us consider the Σ_C-theory T_C, where

- Σ_C consists of two unary function symbols s, p and a constant 0;
- T_C contains all Σ_C-sentences which are true in the structure $(\mathbb{Z}, s, p, 0)$ of the Integers with zero, successor, and predecessor.[23]

Notice that T_C is noetherian, though not locally finite. Indeed, the noetherianity of T_C can be argued from the following arguments: (i) the pure theory of equality over the signature containing a unary function symbol is noetherian (see Appendix A.12); (ii) any extension (over a signature augmented of a finite number of constant symbols) of a noetherian theory remains noetherian; (iii) every Σ_C-formula is T_C-logically equivalent to a $(\Sigma_C \setminus \{p\})$-formula).[24] Moreover, the constraint satisfiability problem of T_C is decidable by quantifier elimination (it is straightforward to adapt the algorithm for the naturals in [18]). T_C can be seen as a 'minimal' theory where to encode an unbounded counter as it is required in order to express

[23] It is possible to use also the structure given by $\mathbb{N}$, 0, successor, and predecessor (the latter is turned into a total function by putting $p(0) := 0$).

[24] In particular, every chain of sets of Σ_C-atoms is T_C-equivalent to a chain of sets of $(\Sigma_C \setminus \{p\})$-atoms. Since this latter has to be eventually constant for logical consequence w.r.t. T_C, so it is the former.

the instructions of the Minsky machines of Table 1. (Below, we abbreviate $\underbrace{s(\cdots(s}_{n \text{ times}}(0)\cdots))$ with the numeral $\overline{n}$.)

We define the totally rigid LTL-theory $\mathcal{T}$ as follows: T_C is underlying theory, there are three system variables $\{a_1, a_2, a_3\}$, and no parameters. Since $\mathcal{T}$ is totally rigid, it is completely determined by its underlying theory, its systems variables, its parameters, and there is no need to specify a rigid subsignature, because all predicate and function symbols are rigid.

We are now in the position to define the encoding of a second reachability problem for a Minsky machine into an LTL-system based on $\mathcal{T}$: we do it for a Minski machine $\mathbf{P}$ and for a configuration (s_0, m_0, n_0) such that $\mathbf{P}$-reachability from (s_0, m_0, n_0) is undecidable.

The transition δ is the disjunction of the following ground sentences:

– for each $\mathbf{P}$-instruction $s \to (t, 1, 0)$ of the first kind, δ contains the disjunct

$$a_1^0 = \overline{s} \wedge a_1^1 = \overline{t} \wedge a_2^1 = s(a_2^0) \wedge a_3^1 = a_3^0;$$

– for each $\mathbf{P}$-instruction $s \to (t, 0, 1)$ of the second kind, δ contains the disjunct

$$a_1^0 = \overline{s} \wedge a_1^1 = \overline{t} \wedge a_2^1 = a_2^0 \wedge a_3^1 = s(a_3^0);$$

– for each $\mathbf{P}$-instruction $s \to (t, -1, 0)[t']$ of the third kind, δ contains the disjuncts

$$\begin{array}{l} \big(a_2^0 \neq 0 \wedge a_1^0 = \overline{s} \wedge a_1^1 = \overline{t} \wedge a_2^1 = p(a_2^0) \wedge a_3^1 = a_3^0\big) \vee \\ \vee \ \big(a_2^0 = 0 \wedge a_1^0 = \overline{s} \wedge a_1^1 = \overline{t'} \wedge a_2^1 = a_2^0 \wedge a_3^1 = a_3^0\big); \end{array}$$

– for each $\mathbf{P}$-instruction $s \to (t, 0, -1)[t']$ of the fourth kind, δ contains the disjuncts

$$\begin{array}{l} \big(a_3^0 \neq 0 \wedge a_1^0 = \overline{s} \wedge a_1^1 = \overline{t} \wedge a_2^1 = a_2^0 \wedge a_3^1 = p(a_3^0)\big) \vee \\ \vee \ \big(a_3^0 = 0 \wedge a_1^0 = \overline{s} \wedge a_1^1 = \overline{t'} \wedge a_2^1 = a_2^0 \wedge a_3^1 = a_3^0\big); \end{array}$$

– finally, δ contains also the idle disjunct

$$a_1^0 = a_1^1 \wedge a_2^1 = a_2^0 \wedge a_3^1 = a_3^0$$

(this disjunct is added in order to make the transition serial).

Let ι be the ground sentence $a_1 = \overline{s_0} \wedge a_2 = \overline{m_0} \wedge a_3 = \overline{n_0}$. We claim that, for a given configurations (s', m', n'), we have that $(s_0, m_0, n_0) \to_{\mathbf{P}}^{\star} (s', m', n')$ iff the formula

$$\Diamond(a_1 = \overline{s'} \wedge a_2 = \overline{m'} \wedge a_3 = \overline{n'})$$

is satisfied in a run of $(\mathcal{T}, \delta, \iota)$. The 'only if' implication of the claim is trivial. For the converse, suppose that there is a run $\mathcal{M}$ of $(\mathcal{T}, \delta, \iota)$ such that

$$\mathcal{M} \models_k a_1 = \overline{s'} \wedge a_2 = \overline{m'} \wedge a_3 = \overline{n'}$$

for some $k \geq 0$. First, notice that one may freely assume that a non-idle disjunct of δ is true in the i-th transition step for $0 \leq i \leq k-1$ (otherwise we can simply remove that step and get a smaller k). Second, as the LTL-theory $\mathcal{T}$ is totally rigid, only the interpretation of the system variables a_1, a_2, a_3 can be different at each time instant - the Σ_C-reduct of the various $\mathcal{M}_i$ being always the same. Such a reduct contains an (elementary) substructure which is isomorphic to the standard model $(\mathbb{Z}, s, p, 0)$ of integers (this is the substructure whose support is the collection of the interpretations of the numerals); moreover, as the system variables take values in the positive subset of that substructure at the initial instant, it is impossible for them to get values outside it for the whole run (to see this, just make an inspection to the definition of the transition δ). This immediately yields $(s_0, m_0, n_0) \rightarrow_{\mathbf{P}}^{\star} (s', m', n')$, as desired. □

4.3 Decidability and Locally Finite LTL-Theories

According to Theorem 4.10, the safety model-checking problem is undecidable in the noetherian compatible case, but we shall prove decidability in the locally finite compatible case.

In the following, let $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ be a locally finite compatible LTL-theory, $(\mathcal{T}, \delta, \iota)$ be an LTL-system specification based on $\mathcal{T}$, and $\upsilon(\underline{a})$ be a ground $\Sigma^{\underline{a},\underline{c}}$-sentence. The related safety model-checking problem amounts to checking whether there exists a run $\mathcal{M} = \{\mathcal{M}_n\}_{n \in \mathbb{N}}$ for $(\mathcal{T}, \delta, \iota)$ such that $\mathcal{M} \models_n \upsilon(\underline{a})$ for some $n \geq 0$: if this is the case, we say that the system is *unsafe* because there is a *bad run of length* n.

We can ignore bad runs of length $n = 0$, because the existence of such runs can be preliminarily decided by checking the ground sentence $\iota(\underline{a}) \wedge \upsilon(\underline{a})$ for T-consistency. So, for $n \geq 1$, taking into account the seriality of the transition, a run of length $n+1$ exists iff the ground $(\bigoplus_{\Sigma_r^c}^{n+2} \Sigma^{\underline{a},\underline{c}})$-sentence

$$\iota^0(\underline{a}^0) \wedge \delta^{0,1}(\underline{a}^0, \underline{a}^1) \wedge \delta^{1,2}(\underline{a}^1, \underline{a}^2) \wedge \cdots \wedge \delta^{n,n+1}(\underline{a}^n, \underline{a}^{n+1}) \wedge \upsilon^{n+1}(\underline{a}^{n+1}) \tag{6}$$

is $\bigoplus_{\Sigma_r^c}^{n+2} T$-satisfiable, where the formulae $\iota^0, \delta^{i,i+1}, \upsilon^{n+1}$ are defined as follows:

- $\iota^0(\underline{a}^0)$ is obtained by replacing each flexible symbol $r \in \Sigma \setminus \Sigma_r$ with r^0 in $\iota(\underline{a})$ (the system variables $\underline{a}$ are similarly renamed as $\underline{a}^0$);
- $\delta^{i,i+1}(\underline{a}^i, \underline{a}^{i+1})$ is obtained by replacing in $\delta(\underline{a}^0, \underline{a}^1)$ the copy r^0 and r^1 of each flexible symbol $r \in \Sigma \setminus \Sigma_r$ with r^i and r^{i+1} respectively (the two copies $\underline{a}^0, \underline{a}^1$ of the system variables $\underline{a}$ are similarly renamed as $\underline{a}^i, \underline{a}^{i+1}$);

– $\upsilon^{n+1}(\underline{a}^{n+1})$ is obtained by replacing each flexible symbol $r \in \Sigma \setminus \Sigma_r$ with r^{n+1} in $\upsilon(\underline{a})$ (the system variables $\underline{a}$ are similarly renamed as $\underline{a}^{n+1}$).

For the sake of simplicity and to improve readability, formula (6) will be written as

$$\iota(\underline{a}^0) \wedge \delta(\underline{a}^0, \underline{a}^1) \wedge \delta(\underline{a}^1, \underline{a}^2) \wedge \cdots \wedge \delta(\underline{a}^n, \underline{a}^{n+1}) \wedge \upsilon(\underline{a}^{n+1}), \tag{7}$$

i.e., only the renaming operation for system variables is explicitly displayed.

Now, for a given $n+1$, an iterated application of Theorem 2.7 (through Proposition A.10) yields the decidability of the satisfiability of formula (7). Unfortunately, this observation is not sufficient to solve the model-checking problem for LTL-system specifications since the length of a run is not known apriori. In the following, we show how to replace the computation of an explicit bound on the length of the runs by a reachability analysis in a finite graph, called the safety graph (thus, the reachability diameter [4] of the safety graph gives also a desired bound, if one wants to determine it). We need some preliminary notions.

Definition 4.11 (Pure formula)**.** A quantifier-free $\Sigma^{\underline{a},\underline{c}} \oplus_{\Sigma^{\underline{c}}_r} \Sigma^{\underline{a},\underline{c}}$-formula δ is said to be

– *purely left* iff for each symbol $r \in \Sigma \setminus \Sigma_r$, we have that r^1 does not occur in δ;

– *purely right* iff for each symbol $r \in \Sigma \setminus \Sigma_r$, we have that r^0 does not occur in δ;

– *pure* iff δ is a boolean combination of purely left and purely right atoms.

First, we transform the ground sentence δ into an existential sentence whose matrix is pure. To this end, it is sufficient to notice that any first-order formula φ is logically equivalent to

$$\exists y\ (y = \varphi_{|p} \wedge \varphi[y]_p), \tag{8}$$

where y is a "fresh" variable (i.e. not occurring in φ) and p is a term position in φ.[25] So, by repeatedly applying (8), we may consider an equivalent formula like

$$\exists \underline{x}\, \tilde{\delta}(\underline{a}^0, \underline{a}^1, \underline{x}), \tag{9}$$

instead of δ, where $\exists \underline{x}$ is a sequence of existential quantifiers and $\tilde{\delta}$ is pure. This preprocessing step is analogous to the purification step of combination procedures [3] - to see the relationship, recall that the signature of δ is the union of $\Sigma^{\underline{a},\underline{c}}$ with a partial copy of itself over the shared signature $\Sigma^{\underline{c}}_r$. Formula (9) is also called the purification of the transition δ and the formula

$$\tilde{\delta}(\underline{a}^0, \underline{a}^1, \underline{d}^0) \tag{10}$$

(where $\underline{d}^0$ are fresh constants replacing the $\underline{x}$) is called the *skolemized purification of the transition* δ.

[25] We use here standard notations from the rewriting literature: $\varphi_{|p}$ is the subterm at position p and $\varphi[y]_p$ is the formula obtained from φ by replacing the subterm at position p by y.

Definition 4.12 ($\tilde{\delta}$-assignment)**.** Let $A_1, \ldots, A_k$ be the atoms occurring in $\tilde{\delta}(\underline{a}^0, \underline{a}^1, \underline{d}^0)$. A *$\tilde{\delta}$-assignment* is a conjunction of the kind

$$B_1 \wedge \cdots \wedge B_k$$

(where B_i is either A_i or $\neg A_i$, for $1 \leq i \leq k$) such that $B_1 \wedge \cdots \wedge B_k \rightarrow \tilde{\delta}$ is a propositional tautology.

In other words, $\tilde{\delta}$-assignments are just boolean assignments satisfying $\tilde{\delta}$; since $\tilde{\delta}$ is pure, we can represent a $\tilde{\delta}$-assignment V in the form $V^l(\underline{a}^0, \underline{a}^1, \underline{d}^0) \wedge V^r(\underline{a}^0, \underline{a}^1, \underline{d}^0)$, where V^l is a purely left conjunction of literals and V^r is a purely right conjunction of literals. As a consequence, a run of length $n+1$ exists iff the ground sentence

$$\iota(\underline{a}^0) \wedge V_1^l(\underline{a}^0, \underline{a}^1, \underline{d}^0) \wedge V_1^r(\underline{a}^0, \underline{a}^1, \underline{d}^0) \wedge \cdots \wedge V_{n+1}^l(\underline{a}^n, \underline{a}^{n+1}, \underline{d}^n) \wedge V_{n+1}^r(\underline{a}^n, \underline{a}^{n+1}, \underline{d}^n) \wedge \upsilon(\underline{a}^{n+1}) \tag{11}$$

is $\bigoplus_{\Sigma_r}^{n+2} T$-satisfiable (here $\underline{d}^0, \underline{d}^1, \ldots, \underline{d}^n$ are $n+1$-copies of the Skolem constants $\underline{d}^0$ and $V_1, \ldots, V_{n+1}$ range over the set of $\tilde{\delta}$-assignments).

We reduce the existence of a satisfiable formula of the kind (11) to a reachability problem in the safety graph defined below. Recall that, since T_r is locally finite, there are finitely many ground $\Sigma_r^{\underline{c}, \underline{a}^0, \underline{a}^1, \underline{d}^0}$-literals which are representative (modulo T_r-equivalence) of all $\Sigma_r^{\underline{c}, \underline{a}^0, \underline{a}^1, \underline{d}^0}$-literals. Furthermore, a guessing $G(\underline{a}^0, \underline{a}^1, \underline{d}^0)$ (in the sense of Definition 3.6) over such literals will be called a *transition Σ_r-guessing*.

Definition 4.13. The *safety graph* associated to the LTL-system specification $(\mathcal{T}, \delta, \iota)$ based on the locally finite compatible LTL-theory $\mathcal{T}$ is the directed graph defined as follows:

- the nodes are the pairs (V, G) where V is a $\tilde{\delta}$-assignment and G is a transition Σ_r-guessing;
- there is an edge $(V, G) \rightarrow (W, H)$ iff the ground sentence

$$G(\underline{a}^0, \underline{a}^1, \underline{d}^0) \wedge V^r(\underline{a}^0, \underline{a}^1, \underline{d}^0) \wedge W^l(\underline{a}^1, \underline{a}^2, \underline{d}^1) \wedge H(\underline{a}^1, \underline{a}^2, \underline{d}^1) \tag{12}$$

  is T-consistent.[26]

[26] Here we still follow our convention of writing only the system variable renamings (flexible symbols being renamed accordingly). In more detail: we make three copies r^0, r^1, r^2 of every flexible symbol $r \in \Sigma \setminus \Sigma_r$. Both V^r and W^l might contain in principle two copies r^0, r^1 of r: the two copies in V^r keep their original names, whereas the two copies in W^l are renamed as r^1, r^2, respectively. However, V^r is a right formula (hence it does not contain r^0) and W^l is a left formula (hence it does not contain r^1): the moral of all this is that only the copy r^1 of r occurs after renaming, which means that (12) is after all just a plain $\Sigma^{\underline{a}^0, \underline{a}^1, \underline{a}^2, \underline{d}^0, \underline{d}^1}$-sentence (thus, it makes sense to test it for T-consistency). Notice that the Skolem constants $\underline{d}^0$ of V^r are renamed as $\underline{d}^1$ in W^l.

The *initial nodes* of the safety graph are the nodes (V, G) such that $\iota(\underline{a}^0) \wedge V^l(\underline{a}^0, \underline{a}^1, \underline{d}^0) \wedge G(\underline{a}^0, \underline{a}^1, \underline{d}^0)$ is T-consistent; the *terminal nodes* of the safety graph are the nodes (V, G) such that $V^r(\underline{a}^0, \underline{a}^1, \underline{d}^0) \wedge \upsilon(\underline{a}^1) \wedge G(\underline{a}^0, \underline{a}^1, \underline{d}^0)$ is T-consistent.

Our decision procedure for the safety model-checking problem relies on the following proposition.

Proposition 4.14. *The system is unsafe iff either $\iota(\underline{a}) \wedge \upsilon(\underline{a})$ is T-satisfiable or there is a path in the safety graph from an initial to a terminal node.*

Proof. Preliminary to the main argument of the proof, which is based on interpolations, let us better analyze the shape of the formula (11) with particular attention to symbols occurring in the various literals. In formula (11), each symbol $r \in \Sigma \setminus \Sigma_r$ can occur in $n+2$-copies $r^0, r^1, \ldots, r^{n+1}$ and the locations of these copies are the following:

(i) r^0 can only occur in $\iota(\underline{a}^0) \wedge V_1^l(\underline{a}^0, \underline{a}^1, \underline{d}^0)$;

(ii) r^i can only occur in $V_i^r(\underline{a}^{i-1}, \underline{a}^i, \underline{d}^{i-1}) \wedge V_{i+1}^l(\underline{a}^i, \underline{a}^{i+1}, \underline{d}^i)$, for $i = 1, \ldots, n$;

(iii) r^{n+1} can only occur in $V_{n+1}^r(\underline{a}^n, \underline{a}^{n+1}, \underline{d}^n) \wedge \upsilon(\underline{a}^{n+1})$.

Now, we are ready to develop the main argument of the proof. Suppose that the system is unsafe. Then, either there is a bad run of length 0 or the formula (11) is satisfiable in a model $\mathcal{N}$ of $\bigoplus_{\Sigma_r}^{n+2} T$ for some $n > 0$. For $i = 0, \ldots, n$, let $G_{i+1}(\underline{a}^0, \underline{a}^1, \underline{d}^0)$ be the Σ_r-transition guessing realized by $(\underline{a}^i, \underline{a}^{i+1}, \underline{d}^i)$ in $\mathcal{N}$ (by this, we mean the set of representative $\Sigma_r^{\underline{c}, \underline{a}^0, \underline{a}^1, \underline{d}^0}$-literals $\psi(\underline{a}^0, \underline{a}^1, \underline{d}^0)$ such that $\mathcal{N} \models \psi(\underline{a}^i, \underline{a}^{i+1}, \underline{d}^i)$). With this choice for the G_i's, the satisfiability of (11) in $\mathcal{N}$ guarantees the existence of the path

$$(V_1, G_1) \rightarrow (V_2, G_2) \rightarrow \cdots \rightarrow (V_{n+1}, G_{n+1}) \tag{13}$$

from the initial node (V_1, G_1) to the terminal node (V_{n+1}, G_{n+1}) within the safety graph.

Viceversa, suppose that there is a path like (13) and that, by contradiction, the system is safe. In particular, this means that the formula

$$\iota(\underline{a}^0) \wedge V_1^l(\underline{a}^0, \underline{a}^1, \underline{d}^0) \wedge V_1^r(\underline{a}^0, \underline{a}^1, \underline{d}^0) \wedge \cdots \wedge V_{n+1}^l(\underline{a}^n, \underline{a}^{n+1}, \underline{d}^n) \wedge V_{n+1}^r(\underline{a}^n, \underline{a}^{n+1}, \underline{d}^n) \wedge \upsilon(\underline{a}^{n+1})$$

is not $\bigoplus_{\Sigma_r}^{n+2} T$-satisfiable. If we apply the interpolation Lemma A.9 to the T_0-compatible theories T and $\bigoplus_{\Sigma_r}^{n+1} T$ (the hypotheses of Lemma A.9 hold by the modularity Proposition A.10), we get a ground $\Sigma_r^{\underline{c}, \underline{a}^0, \underline{a}^1, \underline{d}^0}$-sentence $\psi_1(\underline{a}^0, \underline{a}^1, \underline{d}^0)$ such that

$$T \models \iota(\underline{a}^0) \wedge V_1^l(\underline{a}^0, \underline{a}^1, \underline{d}^0) \rightarrow \psi_1(\underline{a}^0, \underline{a}^1, \underline{d}^0) \tag{14}$$

and such that

$$\psi_1(\underline{a}^0,\underline{a}^1,\underline{d}^0)\wedge V_1^r(\underline{a}^0,\underline{a}^1,\underline{d}^0)\wedge\cdots\wedge V_{n+1}^l(\underline{a}^n,\underline{a}^{n+1},\underline{d}^n)\wedge V_{n+1}^r(\underline{a}^n,\underline{a}^{n+1},\underline{d}^n)\wedge\upsilon(\underline{a}^{n+1}) \quad (15)$$

is not $\bigoplus_{\Sigma_r}^{n+1} T$-satisfiable. Since $G_1(\underline{a}^0,\underline{a}^1,\underline{d}^0)$ is a transition Σ_r-guessing, G_1 represents a maximal choice of representative $\Sigma_r^{\underline{a}^0,\underline{a}^1,\underline{d}^0}$-literals, hence we must have either $T\models G_1\rightarrow\psi_1$ or $T\models G_1\rightarrow\neg\psi_1$ (that is, $T\models\psi_1\rightarrow\neg G$). The latter contradicts (14) and the fact that the node (V_1,G_1) is initial in the safety graph. The former, together with (15) implies that the formula

$$G_1(\underline{a}^0,\underline{a}^1,\underline{d}^0)\wedge V_1^r(\underline{a}^0,\underline{a}^1,\underline{d}^0)\wedge\cdots\wedge V_{n+1}^l(\underline{a}^n,\underline{a}^{n+1},\underline{d}^n)\wedge V_{n+1}^r(\underline{a}^n,\underline{a}^{n+1},\underline{d}^n)\wedge\upsilon(\underline{a}^{n+1}) \quad (16)$$

is not $\bigoplus_{\Sigma_r}^{n+1} T$-satisfiable. We now repeat the argument: we apply the interpolation Lemma A.9 to the T_0-compatible theories T and $\bigoplus_{\Sigma_r}^{n} T$ and we get a ground $\Sigma_r^{\underline{c},\underline{a}^1,\underline{a}^2,\underline{d}^1}$-sentence $\psi_2(\underline{a}^1,\underline{a}^2,\underline{d}^1)$ such that

$$T\models G_1(\underline{a}^0,\underline{a}^1,\underline{d}^0)\wedge V_1^r(\underline{a}^0,\underline{a}^1,\underline{d}^0)\wedge V_2^l(\underline{a}^1,\underline{a}^2,\underline{d}^1)\rightarrow\psi_2(\underline{a}^1,\underline{a}^2,\underline{d}^1) \quad (17)$$

and such that

$$\psi_2(\underline{a}^1,\underline{a}^2,\underline{d}^1)\wedge V_2^r(\underline{a}^1,\underline{a}^2,\underline{d}^1)\wedge\cdots\wedge V_{n+1}^l(\underline{a}^n,\underline{a}^{n+1},\underline{d}^n)\wedge V_{n+1}^r(\underline{a}^n,\underline{a}^{n+1},\underline{d}^n)\wedge\upsilon(\underline{a}^{n+1}) \quad (18)$$

is not $\bigoplus_{\Sigma_r}^{n} T$-satisfiable. Since $G_2(\underline{a}^1,\underline{a}^2,\underline{d}^1)$ is a transition Σ_r-guessing, we must have that either $T\models G_2\rightarrow\psi_2$ or $T\models G_2\rightarrow\neg\psi_2$. The latter contradicts (17) and the existence of an edge $(V_1,G_1)\rightarrow(V_2,G_2)$. The former, together with (18) implies that the formula

$$G_2(\underline{a}^1,\underline{a}^2,\underline{d}^1)\wedge V_2^r(\underline{a}^1,\underline{a}^2,\underline{d}^1)\wedge\cdots\wedge V_{n+1}^l(\underline{a}^n,\underline{a}^{n+1},\underline{d}^n)\wedge V_{n+1}^r(\underline{a}^n,\underline{a}^{n+1},\underline{d}^n)\wedge\upsilon(\underline{a}^{n+1}) \quad (19)$$

is not $\bigoplus_{\Sigma_r}^{n} T$-satisfiable. Continuing in this way, we obtain the T-unsatisfiability of the formula

$$G_{n+1}(\underline{a}^n,\underline{a}^{n+1},\underline{d}^n)\wedge V_{n+1}^r(\underline{a}^n,\underline{a}^{n+1},\underline{d}^n)\wedge\upsilon(\underline{a}^{n+1}) \quad (20)$$

thus contradicting the fact that the node (V_{n+1},G_{n+1}) is final in the safety graph. □

Proposition 4.14 implies the decidability of the safety model-checking problem for locally finite LTL-system specifications.

Theorem 4.15. *The ground safety model-checking problem for a locally finite compatible LTL-system specification is decidable.*

Regarding complexity, the same remarks following the proof of Corollary 3.12 apply here too.

4.4 Examples

In this subsection, we provide examples to which the algorithm suggested by Proposition 4.14 can successfully be applied in order to formally verify safety properties. For the convenience of the reader, we recall the axioms of the theory T_{dlo} of dense linear order since the examples below rely on suitable extensions of it (here and in the following $x < y$ stands for $x \leq y \wedge x \neq y$)

$$\forall x \forall y \forall z \, (x \leq y \wedge y \leq z \rightarrow x \leq y)$$
$$\forall x \forall y \, (x \leq y \vee y \leq x)$$
$$\forall x \forall y \, (x \leq y \wedge y \leq x \rightarrow x = y)$$
$$\forall x \forall y \, (x < y \rightarrow \exists z \, (x < z \wedge z < y))$$

Example 4.16 ([42])**.** Consider a water level controller modeled as follows:

- changes in the water level by inflow/outflow are represented as functions *in* and *out* depending on the water level l and on the time instant; alarm and overflow levels $l_{\text{alarm}} < l_{\text{overflow}}$ are known;
- if the water level l is such that $l \geq l_{\text{alarm}}$ at a given state, then a valve is opened and the water level changes at the next observable time by $l' = in(out(l))$;
- if $l < l_{\text{alarm}}$ then the valve is closed; the water level changes at the next observable time by $l' = in(l)$.

The dependency of the functions *in* and *out* on the time instant means precisely that they can be modeled as *flexible* function symbols depending only on the water level. However, functions *in* and *out* cannot be completely uninterpreted, we impose the following restrictions on them:

$$\forall x \, (x < l_{\text{alarm}} \rightarrow in(x) < l_{\text{overflow}}) \tag{21}$$
$$\forall x \, (x < l_{\text{overflow}} \rightarrow out(x) < l_{\text{alarm}}) \tag{22}$$

Under such restrictions we want to show that from an initial state where $l < l_{\text{alarm}}$ the water level always remains below l_{overflow}.

Let us fix the notation in order to formalize the problem in our framework. We consider the LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$ such that

- $\Sigma = \{in, out, l_{\text{alarm}}, l_{\text{overflow}}, <\}$ where in, out are two unary function symbols, l_{alarm}, l_{overflow} are two constant symbols, $<$ is a binary predicate symbol;
- $\Sigma_r = \{l_{\text{alarm}}, l_{\text{overflow}}, <\}$;

– $T = T_r^\star \cup \{(21), (22)\}$ where $T_r^\star$ is the theory of dense linear orders without endpoints endowed with the further axiom $l_{\text{alarm}} < l_{\text{overflow}}$. In other words, $T_r^\star$ is made of the axioms of T_{dlo} and of the following axioms:

$$\forall x \exists y\ x < y$$
$$\forall x \exists y\ y < x$$
$$l_{\text{alarm}} < l_{\text{overflow}}$$

– l is the only system variable and there are no system parameters (that is, $\underline{a} := \{l\}$ and $\underline{c} := \emptyset$).

It can be shown that the constraint satisfiability problem for T is decidable, that $T_r^\star$ admits quantifier elimination (thus it is the model completion of its universal fragment T_r), and that T_r is effectively locally finite: hence it follows that $\mathcal{T}$ is a locally finite compatible LTL-theory. We consider now the LTL-system specification $(\mathcal{T}, \delta, \iota)$ where δ is

$$\begin{aligned} \delta \quad :\equiv \quad & \big(l_{\text{alarm}} \leq l^0 \rightarrow l^1 = in^0(out^0(l^0))\big) \quad \wedge \\ \wedge \quad & \big(l^0 < l_{\text{alarm}} \rightarrow l^1 = in^0(l^0)\big) \end{aligned}$$

and ι is $l < l_{\text{alarm}}$. Finally, notice that δ is a purely left $(\Sigma^{\underline{a}} \oplus_{\Sigma_r} \Sigma^{\underline{a}})$-formula.

We are interested in the safety model-checking problem in which the unsafe state is described by the formula υ given by $l_{\text{overflow}} < l$. Using the procedure suggested by Theorem 4.15 we can prove that the the system is safe, i.e. that there is no run $\mathcal{M}$ for $(\mathcal{T}, \delta, \iota)$ such that $\mathcal{M} \models \Diamond \upsilon$. We can observe that the task in practice is not extremely hard from a computational point of view, even if, accordingly to Definition 4.13, the graph is made of $2^{32} \times 21$ nodes. In fact, since an edge of the safety graph can connect only T-consistent nodes (i.e, nodes (V, G) such that $V \wedge G$ is T-consistent) and since there are just 50 nodes (modulo T-equivalence) which are T-consistent, at most 50^2 satisfiability tests are required to check whether a terminal node is reachable from an initial one. Moreover, by using suitable heuristics and strategies, the problem becomes computationally even easier: indeed, instead of considering all the edges of the safety graph, it is sufficient to build just the paths starting from the initial nodes or ending in a terminal node (namely applying a forward/backward search strategy). In the former case, it turns out that 26 nodes (modulo T-equivalence) of the safety graph are reachable from an initial node, none of them being a terminal node. In the latter, just 12 nodes are reachable from a terminal node, obviously none of them being an initial node. Hence the dimension of the problem is tractable, although we cannot obviously report full details here.

One might ask if the axioms (21) and (22) are really needed in order to guarantee the safety of the system, or, instead, if it is sufficient to consider just the instantiations of the two axioms above to the water level at the current time. In such a case, T is simply the theory of dense linear order without endpoints endowed with the axiom $l_{\text{alarm}} < l_{\text{overflow}}$; moreover, we have to insert the instances into the transition in such a way they are always satisfied during the flow of time, thus obtaining the new transition

$$\begin{array}{rcll} \delta' & :\equiv & l_{\text{alarm}} \leq l^0 \rightarrow l^1 = in^0(out^0(l^0)) & \wedge \\ & \wedge & l^0 < l_{\text{alarm}} \rightarrow l^1 = in^0(l^0) & \wedge \\ & \wedge & l^0 < l_{\text{alarm}} \rightarrow in^0(l^0) < l_{\text{overflow}} & \wedge \\ & \wedge & l^0 < l_{\text{overflow}} \rightarrow out^0(l^0) < l_{\text{alarm}} & \end{array}$$

In such a system, it is straightforward to see that there is a path into the safety graph from an initial to a terminal node. Consider for example the following path:

$$(V_0, G_0) \longrightarrow (V_1, G_1)$$

where

$$\begin{array}{rcl} V_0(\underline{a}^0, \underline{a}^1) & :\equiv & l^0 < l_{\text{alarm}} \wedge l^0 < l_{\text{overflow}} \wedge l^1 = in^0(out^0(l^0)) \wedge l^1 = in^0(l^0) \wedge \\ & & \qquad \wedge in^0(l^0) < l_{\text{overflow}} \wedge out^0(l^0) < l_{\text{alarm}} \\ G_0(\underline{a}^0, \underline{a}^1) & :\equiv & l^0 < l_{\text{alarm}} < l^1 < l_{\text{overflow}} \end{array}$$

and

$$\begin{array}{rcl} V_1(\underline{a}^0, \underline{a}^1) & :\equiv & l_{\text{alarm}} < l^0 \wedge l^0 < l_{\text{overflow}} \wedge l^1 = in^0(out^0(l^0)) \wedge l^1 = in^0(l^0) \wedge \\ & & \qquad \wedge \neg(in^0(l^0) < l_{\text{overflow}}) \wedge out^0(l^0) < l_{\text{alarm}} \\ G_1(\underline{a}^0, \underline{a}^1) & :\equiv & l_{\text{alarm}} < l^0 < l_{\text{overflow}} < l^1. \end{array}$$

It is easy to check that (V_0, G_0) is an initial node and that (V_1, G_1) is a terminal node; moreover $G_0(\underline{a}^0, \underline{a}^1) \wedge V_1(\underline{a}^1, \underline{a}^2) \wedge G_1(\underline{a}^1, \underline{a}^2)$ is T-consistent (when checking details, remember that our transition δ is a purely left formula).

Example 4.17 (Bakery algorithm)**.** The aim of this example is to use our techniques to analyze the safety of the well-known Lamport's mutual exclusion "Bakery" algorithm. This algorithm can be modeled by a locally finite compatible (and also totally rigid) LTL-system specification in case the number of processors is known.[27] If the number of involved processors

[27] Finite state system specifications are - at least in principle - not enough because the number of tickets is unbounded.

is unknown, we can build for the problem an appropriate $\mathcal{T}$, which is 'almost' a locally finite compatible (not totally rigid anymore) LTL-system specification. We said 'almost' because $\mathcal{T}$ violates our Assumption from Section 4.1.2 in that it has a non-ground transition (some first-order variables are universally quantified in it). We then produce out of $\mathcal{T}$ (by skolemization and instantiation) a locally finite compatible LTL-system specification $\mathcal{T}'$ which is safe iff $\mathcal{T}$ is safe. Safety of $\mathcal{T}'$ can then be easily checked through our techniques. Before analyzing formal details, we point out that the peculiar features of $\mathcal{T}$ that make the whole construction to work are *purely syntactic* in nature and do not need human intervention to be noticed: they basically consist of the finiteness of the set of terms of certain sorts in the skolemized Herbrand universe.

We deal with a sorted language:[28] indeed, we have two sorts, namely P and O. The former is the sort representing the individuals (i.e the involved processes), whereas the latter is used in order to represent tickets. Let us consider the following LTL-theory $\mathcal{T} = \langle \Sigma, T, \Sigma_r, \underline{a}, \underline{c} \rangle$:

- Σ is a sorted signature containing a unary predicate symbol S of sort P, a binary predicate symbol $<: O \times O$, two constant symbols 0 and 1 of sort O, and a unary function symbol $f : P \to O$;
- T axiomatizes, over the sort O, the theory of dense total orders with named distinct endpoints; in other words, T is made of the axioms of T_{dlo} and of the following axioms

$$\forall x\,(0 \leq x)$$
$$\forall x\,(x \leq 1)$$
$$0 < 1$$

  Moreover, the behavior of the function f is constrained by the following further axioms for T:

$$\forall x \forall y\,(f(x) = f(y) \to x = y \vee f(x) = 1) \qquad (f \text{ is "almost-injective"})$$
$$\forall x\,(f(x) = 1 \to \neg S(x))$$

- Σ_r contains the symbols $\{0, 1, <\}$;
- there are no system parameters (i.e. $\underline{c} := \emptyset$) and there is just one system variable t, which is of sort O (i.e. $\underline{a} := \{t\}$).

In order to give an intuitive explanation of what we are modeling, we can think of the values of t at two consecutive instants as the range in which the values of the tickets produced by

[28] There are no problems in extending our results to the many-sorted case.

the "ticket machine" in that interval of time can vary, whereas f can be seen as the function that associates to every individual its current ticket (f is time-dependent, hence flexible, because the ticket is changed after it has been used). We have at our disposal an infinite amounts of tickets whose values are in the interval $[0, 1]$; every individual is inserted into a queue according to the the value of its ticket (the value 1 has the meaning of being out of the queue). Finally, the predicate S models the set of the individuals that are in the critical section.

We leave the reader to check that the constraint satisfiability problem for T is decidable and that T is T_r-compatible for a suitable universal locally finite Σ_r-theory T_r:[29] it follows that $\mathcal{T}$ is a locally finite compatible LTL-theory.

We can associate to $\mathcal{T}$ an LTL-system specification $(\mathcal{T}, \delta, \iota)$ in the following manner: the initial condition is described by the formula

$$\iota \ :\equiv \quad \forall x \, (f(x) = 1) \wedge t = 0,$$

whereas the transition δ is obtained from the conjunction of the following (implicitly universally quantified) formulae:

$$t^0 < t^1 < 1 \tag{23}$$

$$S^0(x) \rightarrow f^1(x) = 1 \tag{24}$$

$$\neg S^0(x) \wedge f^0(x) \neq 1 \rightarrow f^1(x) = f^0(x) \tag{25}$$

$$f^0(x) < f^0(y) \rightarrow \neg S^1(y) \tag{26}$$

$$f^0(x) = 1 \rightarrow f^1(x) = 1 \vee (t^0 \leq f^1(x) \wedge f^1(x) < t^1 \wedge \neg S^1(x)) \tag{27}$$

The meaning of the above formulae is the following:

(23) the range of the values of the tickets produced by the "ticket machine" is strictly increasing during the flow of time;

(24) an individual is removed from the queue immediately after having joined the critical section;

(25) if an individual is in the queue and it is not in the critical section, then its ticket is preserved;

(26) if an individual is not the first in the queue, it cannot enter the critical section;

[29] Take as T_r the theory of linear orders with named distinct endpoints (this admits as a model completion $T_r^\star$, which is the theory of an infinite set over the sort P and of dense linear orders with named distinct endpoints over the sort O).

(27) if an individual is not in the queue, it can remain out of the queue or it can take a ticket (without being immediately served).

The unsafe states are described by the formula

$$\nu \ :\equiv \ \ \exists x \exists y \, (x \neq y \wedge S(x) \wedge S(y)).$$

Since ι, δ, ν all violate our Assumption from Section 4.1.2 because they are not ground, the problem needs to be reformulated (in a *safety/unsafety preserving* way!) in order to become tractable with our techniques.

Consider the LTL-theory $\mathcal{T}' = \langle \Sigma, T, \Sigma_r, \{t\}, \{c_1, c_2\} \rangle$, which is like $\mathcal{T}$ except that two new system parameters c_1, c_2 of sort P have been added. We first skolemize the formula ν into the ground formula

$$\nu' \ :\equiv \ \ c_1 \neq c_2 \wedge S(c_1) \wedge S(c_2),$$

then we instantiate the initial condition ι obtaining

$$\iota' \ :\equiv \ \ t = 0 \wedge f(c_1) = 1 \wedge f(c_2) = 1.$$

Finally we instantiate also the transition δ, thus getting the ground formula δ' which is the conjunctions of (28)-(34) below:[30]

$$t^0 < t^1 < 1 \quad (28)$$

$$(S^0(c_1) \rightarrow f^1(c_1) = 1) \wedge (S^0(c_2) \rightarrow f^1(c_2) = 1) \quad (29)$$

$$(\neg S^0(c_1) \wedge f^0(c_1) \neq 1 \rightarrow f^1(c_1) = f^0(c_1)) \wedge (\neg S^0(c_2) \wedge f^0(c_2) \neq 1 \rightarrow f^1(c_2) = f^0(c_2)) \quad (30)$$

$$f^0(c_1) < f^0(c_2) \rightarrow \neg S^1(c_2) \quad (31)$$

$$f^0(c_2) < f^0(c_1) \rightarrow \neg S^1(c_1) \quad (32)$$

$$f^0(c_1) = 1 \rightarrow f^1(c_1) = 1 \vee (t^0 \leq f^1(c_1) \wedge f^1(c_1) < t^1 \wedge \neg S^1(c_1)) \quad (33)$$

$$f^0(c_2) = 1 \rightarrow f^1(c_2) = 1 \vee (t^0 \leq f^1(c_2) \wedge f^1(c_2) < t^1 \wedge \neg S^1(c_2)) \quad (34)$$

$(\mathcal{T}', \delta', \iota')$ is now an LTL-system specification matching the assumptions of Section 4.1.2; moreover $(\mathcal{T}', \delta', \iota')$ is locally finite compatible for the reasons explained above.

It is not difficult to see that there exists a bad run for $(\mathcal{T}, \iota, \delta)$ (w.r.t. ν) if and only if there exists a bad run for $(\mathcal{T}', \iota', \delta')$ (w.r.t. ν'): the key observation to show this is that one can restrict the interpretation of the sort P in a bad run for $(\mathcal{T}', \iota', \delta')$ so that it consists only on the two individuals c_1, c_2. By applying the algorithm from Proposition 4.14, since $\iota' \wedge \nu'$ is T-inconsistent and since $\delta' \wedge \nu'$ is $(T \oplus_{\Sigma_r} T)$-inconsistent, it follows that $(\mathcal{T}', \iota', \delta')$ is safe w.r.t. ν': consequently, $(\mathcal{T}, \iota, \delta)$ is safe w.r.t. ν too.

[30] Observe that all quantifiers in ι, δ are of sort P and that there are no ground terms in the signature of $\mathcal{T}'$ of that sort, apart from the Skolem constants c_1, c_2. Notice that some instances of δ have been removed, because they are tautological modulo T.

5 Related Work

The high undecidability level of quantified modal logics over the natural numbers flow was realized very early in the sixties by D. Scott; nevertheless, recent literature isolated quite interesting and expressive fragments of quantified LTL which are better behaved from a computational point of view and which can also be decidable in case their extensional part is further restricted to some well-known decidable elementary class: this is the case for instance of the so-called 'monodic fragment' (see [20] for a survey). From another point of view, one can improve the situation by avoiding any interplay between quantifiers and temporal operators [16]; in addition, being especially motivated by verification applications, we were also interested in enriching the extensional part of the language in order to be able to talk about numerical or symbolic data structures. Thus we were naturally lead to consider, from the syntactic point of view, satisfiability of quantifier-free LTL formulae built up from a first-order signature Σ and, from the semantic point of view, we concentrated on LTL constant domain models consisting of a succession $\{\mathcal{M}_i\}_i$ of models of a Σ-theory T. Symbols of Σ and free variables were divided into two groups, the first group being rigidly (i.e. equally) interpreted in all the $\mathcal{M}_i$'s and the second group being possibly differently interpreted in these models. This approach was taken long time ago by [38], who established a decidability result in case the quantifier-free fragment of T is decidable and in case the flexible symbols are free symbols for the theory T (see the Assumption on p.185 of [38]).[31]

By using recent techniques and results from the combination literature, we were able to attack the problem in its full generality and to realize both the undecidability in the unrestricted case (by reduction to combined constraint satisfiability problems for first-order theories) and the decidability within well-known 'combinability' hypotheses [23] for T.

These hypotheses, besides decidability of the universal first-order fragment, were compatibility over a locally finite subtheory in the rigid subsignature (local finiteness was replaced by weaker noetherian requirements in Section 2.2.2).

In the second part of the paper we considered model-checking problems, within the same framework (i.e. under the same hypotheses on T). We got positive decidability results for the safety properties and we plan to extend soon our results to different kinds of properties (liveness, etc.) and finally to the unrestricted LTL model-checking case. Our framework generalizes finite state model-checking for two reasons: first, because the rigid symbols are

[31] Besides a fixed point algorithm, [38] gives also a more simple algorithm on p. 188, lines 15 ff. The latter algorithm consists of a propositional tableaux reduction: while making such a reduction, flexible symbols in the various Hintikka sets are disjointly renamed and a unique T-satisfiability test is performed over the selected candidate regular path (this test involves alien symbols, but decidability is maintained if the additional symbols are all free).

governed by a locally finite theory which is not necessarily an enumerated datatypes theory and secondly because there are no limitations at all on the flexible symbols, whose interpretation is only constrained by the axioms of T.

The literature on infinite state model-checking is extremely vast (e.g., [46, 39, 7, 19] exemplify just some different approaches), we shall make here a comparison only with the literature which is somewhat related to our model-theoretic viewpoint inspired on combination.

The paper [14] makes an extensive review on constrained LTL, which can be seen as a form of model checking for possibly infinite state systems. This form of model-checking does not allow flexible symbols (apart from system variables); moreover specific fixed purely relational structures plays there the role played by the models of the elementary theory T in our approach. Results in [14] are not limited to safety properties; in case our results can be extended beyond safety (as it looks likely to be), *some* of the results in [14] could be seen as specializations of our results to totally rigid system specifications. Other results and techniques from [14] (and also from the recent paper [15]) should nevertheless be seriously taken into account for integration in our settings. A similar observation applies to the rewriting techniques used in [12] in order to obtain decision procedures for interesting (but very special) classes of formulae.

An integration of classic tableaux and automated deduction techniques is presented in [40]. While sharing the goal of combining model-checking algorithms and deductive techniques, [40] provides a uniform framework in which performing such combination with no guarantee on the complete automation of the resulting combination. Similarly, [32] describes a combination of tableaux and automated deduction techniques to automatically solve the model-checking problem of classes of parametrized theories. Although we share some use of tableaux and automated deduction techniques, [32] does not reduce the problem to combination problems in first-order theories.

We discuss the approach in [13] which shares an important distinguishing feature with ours, namely the reduction of safety model-checking problems to satisfiability-modulo-theory constraints. In a sense, our main contribution (Theorem 4.15) identifies precise conditions under which this reduction yields a complete decision procedure (but notice that our safety graph is not just an approximation of the graph of the states of the system, because *pairs* of states are taken into account when building it).

Finally, a long line of research in model-checking infinite-state systems goes under the name of "abstract-check-refine", featuring a combination of finite-state model-checking and decision procedures for first-order theories beginning with the seminal work in [27]. A common feature with our work is emphasis on using decision procedures for the satisfiability problem in first-order theories. However, we are more concerned with precisely characterizing the

termination of the model-checking algorithm while the abstract-check-refine techniques focus on practical usability. Furthermore, for such techniques to scale-up, the decision procedures are required to compute interpolants (see, e.g., [28, 34]) and this may be indeed a difficult task. Instead, our approach should allow one to more easily leverage SMT solvers by designing suitable refinements of the algorithm suggested by Proposition 4.14.

6 Conclusions and Future Work

In this paper, we considered first-order LTL. We studied the decidability of the satisfiability and model-checking problems for various fragments of quantifier-free formulae (modulo a background first-order theory axiomatizing the extensional part of the language). The key technique to obtain our results was a reduction to constraint satisfiability problems in unions of non-disjoint first-order theories: this reduction allowed us to derive undecidability results, but also decidability results through suitable adaptations of extensions of the Nelson-Oppen schema [23, 5]. We also recalled the undecidability of the model-checking problem by a reduction to the reachability problem of Minsky machines [35]. Finally, we gave the decidability of the model-checking problem, when this is restricted to safety properties modulo locally finite and compatible [23] background theories. We also exemplified our techniques on some examples.

There are two main lines of future work. First, we intend to investigate how to exploit SMT solvers to solve model-checking problems with more examples than those considered in this paper. The design of suitable heuristics to efficiently explore the safety graph (cf. Definition 4.13) should be the key to show the viability of our approach. Second, we intend to find termination results for model checking (i) of arbitrary temporal properties (besides those of safety) and (ii) modulo richer background theories (e.g., Presburger Arithmetic). Regarding (i), we envisage to incorporate the Hintikka sets of arbitrary LTL($\Sigma^{\underline{a}}$)-sentences into the safety graph and to extend the algorithm suggested by Proposition 4.14. We believe that (ii) can be achieved by considering transition relations satisfying certain requirements as it is done in, e.g., [15].

References

[1] Franz Baader and Silvio Ghilardi. Connecting many-sorted theories. *Journal of Symbolic Logic*. To appear.

[2] Franz Baader, Silvio Ghilardi, and Cesare Tinelli. A new combination procedure for the word problem that generalizes fusion decidability results in modal logics. *Information and Computation*, 204(10):1413–1452, 2006.

[3] Franz Baader and Cesare Tinelli. Deciding the word problem in the union of equational theories. *Information and Computation*, 178(2):346–390, 2002.

[4] Armin Biere, Alessandro Cimatti, Edmund M. Clarke, Ofer Strichman, and Yunshan Zhu. Bounded model checking. *Advances in Computers*, 58:118–149, 2003.

[5] Maria Paola Bonacina, Silvio Ghilardi, Enrica Nicolini, Silvio Ranise, and Daniele Zucchelli. Decidability and undecidability results for Nelson-Oppen and rewrite-based decision procedures. In U. Furbach and N. Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Computer Science*, pages 513–527, Seattle (WA, USA), 2006. Springer.

[6] Torben Bräuner and Silvio Ghilardi. First-order modal logic. In J. van Benthem, P. Blackburn, and F. Wolter, editors, *Handbook of Modal Logic*, pages 549–620. Elsevier, Amsterdam, 2007.

[7] Olaf Burkart, Didier Caucal, Faron Moller, and Bernhard Steffen. Verification of infinite state structures. In J. A. Bergstra, A. Ponse, and S. A. Smolka, editors, *Handbook of Process Algebras*, pages 545–623. Elsevier, Amsterdam, 2001.

[8] Alexander Chagrov and Michael Zakharyaschev. *Modal Logic*. Clarendon Press, Oxford, 1997.

[9] Chen-Chung Chang and Jerome H. Keisler. *Model Theory*. North-Holland Publishing Co., Amsterdam-London, third edition, 1990.

[10] Alessandro Cimatti, Edmund M. Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. NuSMV 2: An opensource tool for symbolic model checking. In E. Brinksma and K. G. Larsen, editors, *Proceedings of 14th International Conference on Computer Aided Verification (CAV 2002)*, volume 2404 of *Lecture Notes in Computer Science*, pages 359–364, Copenhagen (Denmark), 2002. Springer.

[11] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, 2000.

[12] David Cyrluk and Paliath Narendran. Ground temporal logic: A logic for hardware verification. In D. L. Dill, editor, *Proceedings of the 6th International Conference on Computer Aided Verification (CAV 1994)*, volume 818 of *Lecture Notes in Computer Science*, pages 247–259, Stanford (CA, USA), 1994. Springer-Verlag.

[13] Leonardo de Moura, Harald Rueß, and Maria Sorea. Lazy theorem proving for bounded model checking over infinite domains. In A. Voronkov, editor, *Proceedings of the 18th International Conference on Automated Deduction (CADE 2002)*, volume 2392 of *Lecture Notes in Computer Science*, pages 438–455, Copenhagen (Denmark), 2002. Springer.

[14] Stéphane Demri. Linear-time temporal logics with Presburger constraints: An overview. *Journal of Applied Non-Classical Logics*, 16(3–4):311–347, 2006.

[15] Stéphane Demri, Alain Finkel, Valentin Goranko, and Govert van Drimmelen. Towards a model-checker for counter systems. In S. Graf and W. Zhang, editors, *Proceedings of the 4th International Symposium on Automated Technology for Verification and Analysis (ATVA 2006)*, volume 4218 of *Lecture Notes in Computer Science*, pages 493–507, Beijing (ROC), 2006. Springer.

[16] Alin Deutsch, Liying Sui, and Victor Vianu. Specification and verification of data-driven web services. In A. Deutsch, editor, *Proceedings of the 23rd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2004)*, pages 71–82, Paris (France), 2004. ACM.

[17] Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Mathematical logic*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, second edition, 1994.

[18] Herbert B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, New York-London, 1972.

[19] Javier Esparza, Alain Finkel, and Richard Mayr. On the verification of broadcast protocols. In *Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science (LICS 1999)*, pages 352–359, Trento (Italy), 1999. IEEE Computer Society.

[20] Dov M. Gabbay, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*, volume 148 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam-London, 2003.

[21] Harald Ganzinger. Shostak light. In A. Voronkov, editor, *Proceedings of the 18th International Conference on Automated Deduction (CADE 2002)*, volume 2392 of *Lecture Notes in Computer Science*, pages 332–346, Copenhagen (Denmark), 2002. Springer.

[22] Silvio Ghilardi. Reasoners' cooperation and quantifiers elimination. Technical Report 288-03, Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Milano (Italy), 2003. Available at `http://homes.dsi.unimi.it/~ghilardi`.

[23] Silvio Ghilardi. Model theoretic methods in combined constraint satisfiability. *Journal of Automated Reasoning*, 33(3-4):221–249, 2004.

[24] Silvio Ghilardi, Enrica Nicolini, and Daniele Zucchelli. A comprehensive framework for combined decision procedures. *ACM Transactions on Computational Logic*. To appear. Technical Report version available at `http://homes.dsi.unimi.it/~zucchell/publications/techreport/GhiNiZu-RI304-05.pdf`.

[25] Silvio Ghilardi and Luigi Santocanale. Algebraic and model theoretic techniques for fusion decidability in modal logics. In M. Vardi and A. Voronkov, editors, *Proceedings of the 10th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2003)*, volume 2850 of *Lecture Notes in Computer Science*, pages 152–166, Almaty (Kazakhstan), 2003. Springer.

[26] Rajeev Goré. *Handbook of Tableau Methods*, chapter Tableau Methods for Modal and Temporal Logics, pages 297–396. Kluwer Academic Publishers, 1999.

[27] Susanne Graf and Hassen Saïdi. Verifying invariants using theorem proving. In R. Alur and T. A. Henzinger, editors, *Proceedings of 8th International Conference on Computer Aided Verification (CAV 1996)*, volume 1102 of *Lecture Notes in Computer Science*, pages 196–207, New Brunswick (NJ, USA), 1996. Springer.

[28] Thomas A. Henzinger, Ranjit Jhala, Rupak Majumdar, and Grégoire Sutre. Lazy abstraction. In *Proceedings of the 29th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2002)*, pages 58–70, Portland (OR, USA), 2002. ACM Press.

[29] Gerard J. Holzmann. *The SPIN model checker: Primer and reference manual*. Addison Wesley, 2004.

[30] Ullrich Hustadt and Boris Konev. TRP++: A temporal resolution prover. In M. Baaz, J. Makowsky, and A. Voronkov, editors, *Collegium Logicum*, volume 8, pages 65–79. Kurt Gödel Society, 2004.

[31] Ullrich Hustadt, Boris Konev, and Renate A. Schmidt. Deciding monodic fragments by temporal resolution. In R. Nieuwenhuis, editor, *Proceedings of the 20th International Conference on Automated Deduction (CADE 2005)*, volume 3632 of *Lecture Notes in Computer Science*, pages 204–218, Tallinn (Estonia), 2005. Springer.

[32] Monika Maidl. A unifying model checking approach for safety properties of parameterized systems. In G. Berry, H. Comon, and A. Finkel, editors, *Proceedings of the 13th International Conference on Computer Aided Verification (CAV 2001)*, volume 2102 of *Lecture Notes in Computer Science*, pages 311–323, Paris (France), 2001. Springer.

[33] Zohar Manna and Amir Pnueli. *Temporal Verification of Reactive Systems: Safety.* Springer-Verlag, New York, 1995.

[34] Kenneth L. McMillan. Applications of craig interpolants in model checking. In N. Halbwachs and L. D. Zuck, editors, *Proceedings of the 11th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2005)*, volume 3440 of *Lecture Notes in Computer Science*, pages 1–12, Edinburgh (UK), 2005. Springer.

[35] Marvin L. Minsky. Recursive unsolvability of Post's problem of "tag" and other topics in the theory of Turing machines. *Annals of Mathematics*, 74(3):437–455, 1961.

[36] Greg Nelson and Derek C. Oppen. Simplification by cooperating decision procedures. *ACM Transaction on Programming Languages and Systems*, 1(2):245–257, 1979.

[37] Enrica Nicolini. *Combined decision procedures for constraint satisfiability.* PhD thesis, Dipartimento di Matematica, Università degli Studi di Milano, Milano (Italy), 2007.

[38] David A. Plaisted. A decision procedure for combination of propositional temporal logic and other specialized theories. *Journal of Automated Reasoning*, 2(2):171–190, 1986.

[39] Amir Pnueli, Sitvanit Ruath, and Lenore D. Zuck. Automatic deductive verification with invisible invariants. In T. Margaria and W. Yi, editors, *Proceedings of 7th International Conference in Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2001)*, volume 2031 of *Lecture Notes in Computer Science*, pages 82–97, Genova (Italy), 2001. Springer.

[40] Henny B. Sipma, Tomás E. Uribe, and Zohar Manna. Deductive model checking. *Formal Methods in System Design*, 15(1):49–74, 1999.

[41] A. Prasad Sistla and Edmund M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985.

[42] Viorica Sofronie-Stokkermans. Interpolation in local theory extensions. In U. Furbach and N. Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Computer Science*, pages 235–250, Seattle (WA, USA), 2006. Springer.

[43] Cesare Tinelli. Cooperation of background reasoners in theory reasoning by residue sharing. *Journal of Automated Reasoning*, 3(1):1–31, 2003.

[44] Cesare Tinelli and Mehdi T. Harandi. A new correctness proof of the Nelson-Oppen combination procedure. In F. Baader and K.U. Schulz, editors, *Proceedings of the 1st International Workshop in Frontiers of Combining Systems (FroCoS 1996)*, Applied Logic, pages 103–120, Munich (Germany), 1996. Kluwer Academic Publishers.

[45] Cesare Tinelli and Calogero G. Zarba. Combining non-stably infinite theories. *Journal of Automated Reasoning*, 34(3):209–238, 2005.

[46] Moshe Y. Vardi. Verification of concurrent programs: the automata-theoretic framework. *Annals of Pure and Applied Logic*, 51(1–2):79–98, 1991.

[47] Frank Wolter. Fusions of modal logics revisited. In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic*. CSLI, Stanford (CA, USA), 1998.

A Appendix

This appendix contains some technical facts concerning first-order model theory of classical logic used in completeness proofs for our satisfiability and model-checking algorithms. Most of these facts are straightforward extensions to the case of the combination of infinitely many theories[32] of analogous facts shown in [23] for the case of the combination of two theories. Although one may try to draw these extensions directly from [23] by applying, say, compactness arguments, we give here full proofs from scratch in order to keep this paper completely self-contained.

A.1 More Background

We first recall some further standard background. Given a Σ-structure $\mathcal{M} = (M, \mathcal{I})$ and a subset $C \subseteq M$, the substructure of $\mathcal{M}$ generated by C is the substructure obtained from $\mathcal{M}$ by restricting $\mathcal{I}$ to the subset $\{t^{\mathcal{M}}(\underline{c}) \mid \underline{c} \subseteq C \text{ and } t(\underline{x}) \text{ is a } \Sigma\text{-term}\}$ (here $t^{\mathcal{M}}$ is the function interpreting the term t in $\mathcal{M}$). In case this substructure coincides with $\mathcal{M}$, we say that C is a set of generators for $\mathcal{M}$.

If C is a set of generators for $\mathcal{M}$, the *diagram* $\Delta(\mathcal{M})$ of $\mathcal{M}$ (w.r.t. Σ, C) consists of all ground Σ^C-literals that hold in $\mathcal{M}$; analogously, the *elementary diagram* $\Delta^e(\mathcal{M})$ of $\mathcal{M}$ (w.r.t. Σ, C) consists of all ground Σ^C-sentences that hold in $\mathcal{M}$ (often C is not specified at all, in these cases it is assumed to coincide with the whole carrier set of $\mathcal{M}$).

Diagrams (in combination with the compactness of the logical consequence relation) will be repeatedly used. A typical standard use is the following: suppose that we want to embed $\mathcal{M}$ into a model of a theory T, then it is sufficient to check that $T \cup \Delta(\mathcal{M})$ is consistent. This argument is justified by Robinson's Diagram Lemma [9], which relates embeddings and diagrams as follows.

Lemma A.1 (Robinson's Diagram Lemma)**.** *Let $\mathcal{M}$ be a Σ-structure generated by a set C, and let $\mathcal{N}$ be another Σ-structure; then $\mathcal{M}$ can be embedded (resp. elementarily embedded) into $\mathcal{N}$ iff $\mathcal{N}$ can be expanded to Σ^C-model of the diagram $\Delta(\mathcal{M})$ (resp. of the elementary diagram $\Delta^e(\mathcal{M})$) of $\mathcal{M}$ w.r.t. Σ, C.*

The technique used for proving Lemma A.1 is simple, we sketch it. If we have an expansion of $\mathcal{N}$ to a Σ^C-structure (to be called $\mathcal{N}$ again for simplicity), then, since every element of the support of $\mathcal{M}$ is of the kind $t^{\mathcal{M}}(\underline{c})$ for some $c \subseteq C$, we can define the embedding μ by putting $\mu(t^{\mathcal{M}}(\underline{c})) := t^{\mathcal{N}}(\underline{c}^{\mathcal{N}})$: this is well-defined and it is an embedding precisely because

[32] More precisely, the relevant case is the case of the combination of countably many partially renamed copies of the same theory.

$\mathcal{N} \models \Delta(\mathcal{M})$. Conversely, if we have the embedding μ, then we can get the desired expansion by taking $c^{\mathcal{N}} := \mu(c)$ for all $c \in C$.

Since a surjective embedding is just an isomorphism, the argument just sketched shows also the following fact:

Lemma A.2. *If two Σ-structures $\mathcal{M}$, $\mathcal{N}$ are both generated by a set C and if one of them, say $\mathcal{N}$, satisfies the other's diagram (w.r.t. Σ, C), then the two structures are Σ^C-isomorphic.*

Ground formulae are invariant under embeddings in the following sense.

Lemma A.3. *Let $\mathcal{M} = (M, \mathcal{I})$ be a Σ-structure that can be embedded into another Σ-structure $\mathcal{N}$. For all ground Σ^M-sentences φ, we have that*

$$\mathcal{M} \models \varphi \qquad \Leftrightarrow \qquad \mathcal{N} \models \varphi,$$

where $\mathcal{N}$ is extended to a Σ^M-structure by interpreting each $a \in M$ by its image under the embedding.

Next lemma states the well-known property (called submodel-completeness) of theories enjoying quantifier-elimination:

Lemma A.4. *Suppose that $T^\star$ is a Σ_r-theory enjoying quantifier elimination and that Δ is a diagram of a substructure $\mathcal{R} = (R, \mathcal{J})$ of a model $\mathcal{M}$ of $T^\star$; then the Σ^R-theory $T^\star \cup \Delta$ is complete.*

Proof. By Robinson Diagram Lemma A.1, the models of $T^\star \cup \Delta$ are the models of $T^\star$ endowed with a Σ_r-embedding from $\mathcal{R}$. One such model is $\mathcal{M}$; we show that any other model $\mathcal{M}'$ satisfies the same Σ^R-sentences as $\mathcal{M}$ (we assume without loss of generality the Σ_r-embedding from $\mathcal{R}$ into $\mathcal{M}'$ to be an inclusion). Pick an arbitrary Σ^R-sentence $\varphi(\underline{c})$ (where the $\underline{c}$ are parameters from the set of generators of $\mathcal{R}$ used in order to build Δ): this sentence is equivalent, modulo $T^\star$, to a ground Σ^R-sentence $\varphi^\star(\underline{c})$. Since truth of ground sentences is preserved by substructures (see Lemma A.3), we have the following chain of equivalences

$$\mathcal{M}' \models \varphi(\underline{c}) \quad \Leftrightarrow \quad \mathcal{M}' \models \varphi^\star(\underline{c}) \quad \Leftrightarrow \quad \mathcal{R} \models \varphi^\star(\underline{c}) \quad \Leftrightarrow \quad \mathcal{M} \models \varphi^\star(\underline{c}) \quad \Leftrightarrow \quad \mathcal{M} \models \varphi(\underline{c}),$$

showing our claim. □

Next result is also part of basic classical model theory: a proof of it can be easily deduced from Craig's Interpolation Theorem (alternatively, a direct proof using a double chain argument is possible, see [9], pp. 141-142):

Theorem A.5 (Robinson's Joint Consistency Theorem)**.** *Let H_1, H_2 be, respectively, consistent Θ_1, Θ_2-theories and let Θ_0 be the signature $\Theta_1 \cap \Theta_2$. Suppose that there is a complete Θ_0-theory H_0 such that $H_0 \subseteq H_1$ and $H_0 \subseteq H_2$; then $H_1 \cup H_2$ is a consistent $\Theta_1 \cup \Theta_2$-theory.*

A.2 Structure Amalgamations

The statement of next Lemma extends the statement of Lemma 9.3 from [23] (and is proved in the same way):

Lemma A.6. *Let T_i be Σ_i-theories (for $i \in I$) and let Σ_r be a subsignature of all the Σ_i's. Let*

$$\Gamma_1, \ldots, \Gamma_i, \ldots \quad (i \in I)$$

be sets of ground $\Sigma_i^{\underline{a}_i, \underline{c}}$-clauses (here $\underline{a}_i, \underline{c}$ are free constants); a set $\mathcal{B}$ of positive ground $\Sigma_r^{\underline{c}}$-clauses is said to be saturated iff for every $i \in I$ and for every positive ground $\Sigma_r^{\underline{c}}$-clause C it happens that:

$$T_i \cup \Gamma_i \cup \mathcal{B} \models C \quad \Rightarrow \quad C \in \mathcal{B}.$$

Suppose now that $\mathcal{B}$ is saturated and does not contain the empty clause. Then there are $\Sigma_i^{\underline{a}_i, \underline{c}}$-structures $\mathcal{M}_i$ such that $\mathcal{M}_i \models T_i \cup \Gamma_i \cup \mathcal{B}$; moreover, the $\Sigma_r^{\underline{c}}$-substructures generated by the elements (denoted by) $\underline{c}$ coincide for all the $\mathcal{M}_i$'s.

Proof. A set of ground $\Sigma_r^{\underline{c}}$-literals is said to be exhaustive iff it contains, for every ground $\Sigma_r^{\underline{c}}$-literal A, either A itself or its negation. The statement of the lemma is proved if we are able to find an exhaustive set Δ of ground $\Sigma_r^{\underline{c}}$-literals which is consistent with $T_i \cup \Gamma_i \cup \mathcal{B}$ for each $i \in I$. In this case, in fact, given models $\mathcal{M}_i \models T_i \cup \Gamma_i \cup \mathcal{B} \cup \Delta$, we have that the $\Sigma_r^{\underline{c}}$-substructures generated by $\underline{c}$ in all the $\mathcal{M}_i$'s all have diagram Δ, consequently they are $\Sigma_r^{\underline{c}}$-isomorphic (and can be made coincident by suitable renaming).

We shall adapt the notion of productive clause used in nowadays refutational completeness proofs for e.g. resolution or paramodulation based calculi. Consider any strict total terminating order on ground $\Sigma_r^{\underline{c}}$-atoms and extend it to a strict total terminating order $>$ for positive ground $\Sigma_r^{\underline{c}}$-clauses by taking standard multiset extension. We shall define increasing sets Δ_C^+ (varying $C \in \mathcal{B}$) of ground $\Sigma_r^{\underline{c}}$-atoms as follows. Recall that, as the empty clause is not in $\mathcal{B}$, all positive clauses in $\mathcal{B}$ are of the kind $A \vee A_1 \vee \cdots \vee A_n$ ($n \geq 0$).

The definition is by transfinite induction on $>$. Say that the clause $C \equiv A \vee A_1 \vee \cdots \vee A_n$ from $\mathcal{B}$ is *productive* iff (i) $\{A\} > \{A_1, \ldots, A_n\}$ and (ii) $A_1, \ldots, A_n \notin \Delta_{<C}^+$ (where $\Delta_{<C}^+$ is $\bigcup_{D<C} \Delta_D^+$). Now, if C is productive, we let Δ_C^+ to be $\Delta_{<C}^+ \cup \{A\}$, otherwise Δ_C^+ is simply $\Delta_{<C}^+$.

Let Δ^+ be $\bigcup_{C \in \mathcal{B}} \Delta_C^+$ and Δ be $\Delta^+ \cup \{\neg A \mid A$ is a ground $\Sigma_r^{\underline{c}}$-atom not belonging to $\Delta^+\}$. By construction, $\Delta \models \mathcal{B}$, so we simply need to show that $T_i \cup \Gamma_i \cup \Delta$ is consistent for each $i \in I$. We need a preliminary claim.

Claim: if the clause $A \vee A_1 \vee \cdots \vee A_n$ is productive and A is the maximum atom in it, then $A_1, \ldots, A_n \notin \Delta^+$: this is evident, as the A_i's could only be produced by clauses smaller than $A \vee A_1 \vee \cdots \vee A_n$.

Suppose now that $T_i \cup \Gamma_i \cup \Delta$ is not consistent. Then there are ground atoms $B_1, \ldots, B_m \notin \Delta^+$ and productive clauses

$$\begin{aligned} C_1 &\equiv A_1 \vee A_{11} \vee \cdots \vee A_{1k_1} \\ &\cdots \\ C_n &\equiv A_n \vee A_{n1} \vee \cdots \vee A_{nk_n} \end{aligned}$$

(with maximum atoms $A_1, \ldots, A_n$, respectively), such that

$$T_i \cup \Gamma_i \cup \{A_1, \ldots, A_n\} \models B_1 \vee \cdots \vee B_m.$$

By trivial logical manipulations, it follows that

$$T_i \cup \Gamma_i \cup \{C_1, \ldots, C_n\} \models \bigvee_{i,j} A_{ij} \vee B_1 \vee \cdots \vee B_m.$$

As $C_1, \ldots, C_n$ are clauses in $\mathcal{B}$ and as $\mathcal{B}$ is saturated, the clause

$$D \equiv \bigvee_{i,j} A_{ij} \vee B_1 \vee \cdots \vee B_m$$

is also in $\mathcal{B}$. By construction (anyway, either D is productive or not) some of the atoms $\{A_{11}, \ldots, A_{nk_n}, B_1, \ldots, B_m\}$ are in Δ^+. By the claim, $A_{11}, \ldots, A_{nk_n}$ cannot be there, so one of the B_j's is in Δ^+, contradiction. □

Next Lemma also extends a fact (namely Lemma 9.4) established in [23]:

Lemma A.7. *Let $\Sigma_i^{\underline{c},\underline{a}_i}$ (for $i \in I$) be signatures (expanded with free constants $\underline{c}, \underline{a}_i$), whose pairwise intersections are all equal to a certain signature $\Sigma_r^{\underline{c}}$ (that is, we have $\Sigma_i^{\underline{c},\underline{a}_i} \cap \Sigma_j^{\underline{c},\underline{a}_j} = \Sigma_r^{\underline{c}}$ for all distinct $i, j \in I$). Suppose we are also given Σ_i-theories T_i which are all T_r-compatible, where $T_r \subseteq \bigcap_i T_i$ is a universal Σ_r-theory; let finally $\{\mathcal{M}_i = (M_i, \mathcal{I}_i)\}_{i \in I}$ be a sequence of $\Sigma_i^{\underline{c},\underline{a}_i}$-structures which are models of T_i and satisfy the same $\Sigma_r^{\underline{c}}$-atoms. In these hypotheses, there exist a $\bigcup_i(\Sigma_i^{\underline{c},\underline{a}_i})$-structure $\mathcal{M} \models \bigcup_i T_i$ such that for each i, $\mathcal{M}_i$ has a $\Sigma_i^{\underline{c},\underline{a}_i}$-embedding into $\mathcal{M}$.*

Proof. By Robinson Diagram Lemma A.1 and by Lemma A.2 (and up to a partial renaming of the support sets), the fact that the $\mathcal{M}_i$ satisfy the same $\Sigma_r^{\underline{c}}$-atoms is another way of saying that they share the same $\Sigma_r^{\underline{c}}$-substructure generated by the $\underline{c}$ (let us call $\mathcal{R} = (R, \mathcal{J})$ this substructure); by T_r-compatibility, we may also freely assume that $\mathcal{M}_i \models T_i \cup T_r^\star$. Notice

also that, by Lemma A.4 above, the theory $T_r^\star \cup \Delta$ is complete, where Δ is the diagram of $\mathcal{R}$ as a Σ_r-structure.

Again by Robinson Diagram Lemma, we only need to show that the union of the elementary diagrams $\Delta_i^e(\mathcal{M}_i)$ is consistent:[33] here $\Delta_i^e(\mathcal{M}_i)$ is the elementary diagram of $\mathcal{M}_i$ as a $\Sigma_i^{\underline{c},\underline{a}_i}$-structure.

By compactness, we can freely assume that the index set I is finite, let it be $\{1,\dots,k\}$ and let us argue by induction on k. For $k=1$, there is nothing to prove and for $k>1$, we use Robinson's Joint Consistency Theorem as follows.

By renaming some elements in the supports if needed, we can freely suppose that the sets $M_1 \setminus R$ and $(M_2 \cup \dots \cup M_k) \setminus R$ are disjoint. Given the hypotheses of the Lemma on the signatures $\Sigma_i^{\underline{c},\underline{a}_i}$, we can apply the Joint Consistency Theorem to the theories $\Delta^e(\mathcal{M}_1)$ and $\Delta^e(\mathcal{M}_2) \cup \dots \cup \Delta^e(\mathcal{M}_k)$: in fact, they are both consistent (the latter by induction) and their both contain the complete subtheory $T_r^\star \cup \Delta$ in the shared subsignature. This proves that $\Delta^e(\mathcal{M}_1) \cup \dots \cup \Delta^e(\mathcal{M}_k)$ is consistent, as desired. □

If we put together the two previous lemmas, we get the following fact:

Lemma A.8. *Suppose we are given the following data:*

(i) *I is a (possibly infinite) set of indexes;*

(ii) *$\Sigma_i^{\underline{c},\underline{a}_i}$ (for $i \in I$) are signatures (expanded with free constants $\underline{c}, \underline{a}_i$), whose pairwise intersections are all equal to a certain signature $\Sigma_r^{\underline{c}}$ (that is, we have $\Sigma_i^{\underline{c},\underline{a}_i} \cap \Sigma_j^{\underline{c},\underline{a}_j} = \Sigma_r^{\underline{c}}$ for all distinct $i,j \in I$);*

(iii) *T_i are Σ_i-theories (for $i \in I$) which are all T_r-compatible, where $T_r \subseteq \bigcap_i T_i$ is a universal Σ_r-theory;*

(iv) *$\{\Gamma_i\}_{i\in I}$ are sets of ground $\Sigma_i^{\underline{a}_i,\underline{c}}$-clauses;*

(v) *$\mathcal{B}$ is a set of positive ground $\Sigma_r^{\underline{c}}$-clauses not containing the empty clause and satisfying the following condition for every $i \in I$ and for every positive ground $\Sigma_r^{\underline{c}}$-clause C:*

$$T_i \cup \Gamma_i \cup \mathcal{B} \models C \quad \Rightarrow \quad C \in \mathcal{B}.$$

If the above data are given, then there exists a $\bigcup_i(\Sigma_i^{\underline{c},\underline{a}_i})$-structure $\mathcal{M} \models \bigcup_i(T_i \cup \Gamma_i)$. Equivalently: there exist $\Sigma_i^{\underline{c},\underline{a}_i}$-structures $\mathcal{M}_i$ ($i \in I$) satisfying $T_i \cup \Gamma_i$, whose $\Sigma_r^{\underline{c}}$-reducts coincide.

Next Lemma is a variant of Theorem 5.2 from [23] (but the proof below is different):

[33] We need the elementary diagrams here, and not just diagrams, because we want the model to be built to be a model of $\bigcup_i T_i$.

Lemma A.9. *Suppose that T_0, T_1, T_2 are $\Sigma_0, \Sigma_1, \Sigma_2$-theories (respectively) such that $\Sigma_0 = \Sigma_1 \cap \Sigma_2$, T_1 is T_0-compatible, and T_2 is T_0-compatible; if the ground $\Sigma_1^{\underline{a},\underline{b}}$-sentence $\psi_1(\underline{a},\underline{b})$ and the ground $\Sigma_2^{\underline{b},\underline{c}}$-sentence $\psi_2(\underline{b},\underline{c})$ (here the tuples of free constants $\underline{a},\underline{b},\underline{c}$ are pairwise disjoint) are such that $\psi_1(\underline{a},\underline{b}) \wedge \psi_2(\underline{b},\underline{c})$ is $T_1 \cup T_2$-inconsistent, then there is a ground $\Sigma_0^{\underline{b}}$-sentence $\psi_0(\underline{b})$ such that $T_1 \models \psi_1(\underline{a},\underline{b}) \rightarrow \psi_0(\underline{b})$ and $T_2 \models \psi_0(\underline{b}) \rightarrow \neg\psi_2(\underline{b},\underline{c})$.*

Proof. By compactness, it is sufficient to show that the set Ψ of ground $\Sigma_0^{\underline{b}}$-sentences $\psi_0(\underline{b})$ such that $T_1 \models \psi_1(\underline{a},\underline{b}) \rightarrow \psi_0(\underline{b})$ is not T_2-consistent with $\psi_2(\underline{b},\underline{c})$. Suppose it is, hence there is a T_2-model $\mathcal{M}_2$ of $\Psi \cup \{\psi_2(\underline{b},\underline{c})\}$. Let $\mathcal{R}$ be the Σ_0-substructure of $\mathcal{M}$ generated by the $\underline{b}$'s and let Δ be its diagram. We claim that Δ is T_1-consistent with $\psi_1(\underline{a},\underline{b})$: this is because, if $\psi_0(\underline{b})$ is a ground $\Sigma_0^{\underline{b}}$-sentence true in $\mathcal{R}$ and not consistent with $\psi_1(\underline{a},\underline{b})$, then $\neg\psi_0(\underline{b})$ would be in Ψ and hence would be true in $\mathcal{R}$, contradiction. Since Δ is T_1-consistent with $\psi_1(\underline{a},\underline{b})$, there is a model $\mathcal{M}_1$ of T_1 (having $\mathcal{R}$ as a substructure) in which $\psi_1(\underline{a},\underline{b})$ is true. By Lemma A.7 (take $I = \{1,2\}$), the models $\mathcal{M}_1, \mathcal{M}_2$ embeds over $\mathcal{R}$ into a model $\mathcal{M}$ of $T_1 \cup T_2$; but then $\mathcal{M}$ is also a model of $\psi_1(\underline{a},\underline{b}) \wedge \psi_2(\underline{b},\underline{c})$ (because $\psi_1(\underline{a},\underline{b})$ and $\psi_2(\underline{b},\underline{c})$ are ground, see Lemma A.3), a contradiction. □

Proposition A.10. *If T_0, T_1, T_2 are $\Sigma_0, \Sigma_1, \Sigma_2$-theories (respectively) such that $\Sigma_0 = \Sigma_1 \cap \Sigma_2$, T_1 is T_0-compatible, and T_2 is T_0-compatible, then $T_1 \cup T_2$ is T_0-compatible too.*

Proof. This is Proposition 4.4 from [23]: we report the proof here. Take a model $\mathcal{M} = (M, \mathcal{I})$ of $T_1 \cup T_2$ and embeds its Σ_i-reducts into models $\mathcal{M}_i = (M_i, \mathcal{I}_i)$ of $T_i \cup T_0^\star$ $(i = 1, 2)$. We can freely suppose that the embeddings are inclusions and that we have $M = M_1 \cap M_2$ for supports. Now $T_0^\star \cup \Delta(\mathcal{M})$ is a complete theory by Lemma A.4 (here $\Delta(\mathcal{M})$ is the diagram of $\mathcal{M}$ as a Σ_0-structure), hence by Robinson Joint Consistency Theorem A.5 there is a model $\mathcal{N} = (N, \mathcal{J})$ of $\Delta^e(\mathcal{M}_1) \cup \Delta^e(\mathcal{M}_2)$. It follows that $\mathcal{N}$ is a $(\Sigma_1 \cup \Sigma_2)^{M_1 \cup M_2}$-model of $T_1 \cup T_2 \cup T_0^\star$ and that there are Σ_i^M-embeddings $\mu_i : \mathcal{M}_i \longrightarrow \mathcal{N}$. In particular, for $b \in M$, we have $\mu_1(b) = b^{\mathcal{N}} = \mu_2(b)$; let us call μ the common restriction of μ_1 and μ_2 to M. We show that μ is a $(\Sigma_1 \cup \Sigma_2)$-embedding of $\mathcal{M}$ into $\mathcal{N}$. Observe in fact that for every n-ary Σ_i-function symbol f and for every n-tuple $\underline{b}$ of elements from the support of $\mathcal{M}$, we have[34]

$$\mu(f^{\mathcal{M}}(\underline{b})) = \mu_i(f^{\mathcal{M}_i}(\underline{b})) = f^{\mathcal{N}}(\mu_i(\underline{b})) = f^{\mathcal{N}}(\mu(\underline{b}));$$

analogously, for every n-ary Σ_i-predicate symbol P, we have

$$\mathcal{M} \models P(\underline{b}) \text{ iff } \mathcal{M}_i \models P(\underline{b}) \text{ iff } \mathcal{N} \models P(\mu_i(\underline{b})) \text{ iff } \mathcal{N} \models P(\mu(\underline{b})).$$

This proves that $\mu : \mathcal{M} \longrightarrow \mathcal{N}$ is a $(\Sigma_1 \cup \Sigma_2)$-embedding. □

[34] Here, if $\underline{b} = (b_1, \ldots, b_n)$, we write e.g. $\mu(\underline{b})$ for the tuple $(\mu(b_1), \ldots, \mu(b_n))$.

A.3 More on Noetherian Theories

We conclude this Appendix by giving the proof of a couple of statements concerning noetherian theories. The following lemma transfers the termination property from sets of atoms to sets of positive clauses:

Lemma A.11. *Every infinite ascending chain of sets of positive ground Σ^c_r-clauses is eventually constant for logical consequence modulo a noetherian Σ-theory T_r.*

Proof. By contradiction, suppose not; in this case it is immediate to see that there are infinitely many positive ground T_r-clauses $C_1, C_2, \ldots$ such that for all i the clause C_i is not a logical consequence of $T_r \cup \{C_1, \ldots, C_{i-1}\}$.

Let us build a chain of trees $\mathcal{T}_0 \subseteq \mathcal{T}_1 \subseteq \mathcal{T}_2 \subseteq \cdots$, whose nodes are labeled by positive ground Σ^c_r-atoms as follows. $\mathcal{T}_0$ consists of the root only, which is labeled $\top$. Suppose $\mathcal{T}_{i-1}$ is already built and consider the clause $C_i \equiv B_1 \vee \cdots \vee B_m$. To build $\mathcal{T}_i$, do the following for every leaf K of $\mathcal{T}_{i-1}$ (let the branch leading to K be labeled by $A_1, \ldots, A_k$): append new sons to K labeled $B_1, \ldots, B_m$, respectively, if C_i is such that $T_r \cup \{A_1, \ldots, A_k\} \not\models C_i$ (if this is not the case, do nothing for the leaf K).

Consider now the union tree $\mathcal{T} = \bigcup \mathcal{T}_i$: since, whenever a node labeled A_{k+1} is added, A_{k+1} is not a logical consequence w.r.t. T_r of the formulae labeling the predecessor nodes, by the noetherianity of T_r all branches are then finite and by König lemma the whole tree is itself finite. This means that for some index j, the examination of clauses C_i (for $i > j$) did not yield any modification of the already built tree. Now, C_{j+1} is not a logical consequence of $T_r \cup \{C_1, \ldots, C_j\}$: this means that there is a Σ^c_r-structure $\mathcal{M}$ which is a model of T_r and in which all atoms of C_{j+1} are false and the $C_1, \ldots, C_j$ are all true. By induction on $i = 0, \ldots, j$, it is easily seen that there is a branch in $\mathcal{T}_i$ whose labeling atoms are true in $\mathcal{M}$: this contradicts the fact that the tree $\mathcal{T}_j$ has not been modified in step $j + 1$. □

Proposition A.12. *The empty theory T over the signature Σ containing only the unary function symbol f is noetherian.*

Proof. By contradiction, suppose that there is a chain $\Theta_1 \subseteq \Theta_2 \subseteq \cdots \subseteq \Theta_n \subseteq \cdots$ of sets of ground $\Sigma^{\underline{a}}$-atoms which is not eventually constant for logical consequence w.r.t. T. Without loss of generality, we can assume that $\Theta_1 \subseteq \Theta_2 \subseteq \cdots \subseteq \Theta_n \subseteq \cdots$ is such that for each i there exists a $\Sigma^{\underline{a}}$-atom ℓ_i such that $T \cup \Theta_{i-1} \not\models \ell_i$.

Notice that, since f is a unary function symbol, each element of the infinite sequence $\{\ell_i\}_{i \in \mathbb{N}}$ of $\Sigma^{\underline{a}}$-atoms is a Σ^{a_i, a_j}-atom (for some $a_i, a_j \in \underline{a}$). Thus, since $\underline{a}$ is finite, we can extract an infinite subsequence of ground $\Sigma^{a,b}$-atoms (for some fixed elements $a, b \in \underline{a}$) inducing an infinite ascending chain $\Theta_{1|\Sigma^{a,b}} \subseteq \Theta_{2|\Sigma^{a,b}} \subseteq \cdots \subseteq \Theta_{n|\Sigma^{a,b}} \subseteq \cdots$ which is not eventually

constant for logical consequence w.r.t. T (here $\Theta_{i|\Sigma}$ is the collection of all the ground Σ-atoms occurring in Θ_i).

Suppose that a $\Sigma^{a,b}$-atom of the kind $\ell := f^m(a) = f^n(a)$ occurs in such an infinite subsequence (here $m \neq n$ otherwise $T \models \ell$, contrary to our choice of these atoms). Notice that $T \cup \ell$ is such that there are only finitely many Σ^a-terms that are not logically equivalent w.r.t. $T \cup \ell$, which implies that every infinite ascending chain of sets of ground Σ^a-atoms is eventually constant for logical consequence w.r.t. $T \cup \ell$ (the same argument apply to atoms of the kind $\ell := f^m(b) = f^n(b)$).

Suppose now that a $\Sigma^{a,b}$-atom of the kind $\ell := f^m(a) = f^n(b)$ belongs to such an infinite chain of $\Sigma^{a,b}$-atoms. The only $\Sigma^{a,b}$-atoms of the form $f^{m'}(a) = f^{n'}(b)$ not implied by $T \cup \ell$ are such that either (i) $m - n \neq m' - n'$ or (ii) $m' < m$ and $n' < n$. It is clear that there are only finitely many atoms of the kind (ii); for (i), notice that $f^m(a) = f^n(b) \wedge f^{m'}(a) = f^{n'}(b)$ implies that $f^{m+n'}(a) = f^{n+n'}(b) = f^{m'+n}(a)$ and that $f^{n+m'}(b) = f^{m+m'}(a) = f^{n'+m}(b)$ (where $m + n' \neq m' + n$ by (i)), so we are reduced to the first case.

The arguments above imply that the chain $\Theta_{1|\Sigma^{a,b}} \subseteq \Theta_{2|\Sigma^{a,b}} \subseteq \cdots \subseteq \Theta_{n|\Sigma^{a,b}} \subseteq \cdots$ is eventually constant for logical consequence w.r.t. T. Contradiction. □