

# Decidable Fragments of $LTL_f$ Modulo Theories

Luca Geatti<sup>a</sup>, Alessandro Gianola<sup>b</sup>, Nicola Gigante<sup>b</sup> and Sarah Winkler<sup>b</sup>

<sup>a</sup>University of Udine, Italy

<sup>b</sup>Free University of Bozen-Bolzano, Italy

ORCID ID: Luca Geatti <https://orcid.org/0000-0002-7125-787X>,

Alessandro Gianola <https://orcid.org/0000-0003-4216-5199>,

Nicola Gigante <https://orcid.org/0000-0002-2254-4821>, Sarah Winkler <https://orcid.org/0000-0001-8114-3107>

**Abstract.** We study Linear Temporal Logic Modulo Theories over Finite Traces ( $LTL_f^{MT}$ ), a recently introduced extension of LTL over finite traces ( $LTL_f$ ) where propositions are replaced by first-order formulas and where first-order variables referring to different time points can be compared. In general,  $LTL_f^{MT}$  was shown to be semi-decidable for any decidable first-order theory (e.g., linear arithmetic), with a tableau-based semi-decision procedure.

In this paper we present a sound and complete pruning rule for the  $LTL_f^{MT}$  tableau. We show that for any  $LTL_f^{MT}$  formula that satisfies an abstract, semantic condition, that we call finite memory, the tableau augmented with the new rule is also guaranteed to terminate. Last but not least, this technique allows us to establish novel decidability results for the satisfiability of several fragments of  $LTL_f^{MT}$ , as well as to give new decidability proofs for classes that are already known.

## 1 Introduction

Linear Temporal Logic (LTL) [36] and its finite-traces counterpart ( $LTL_f$ ) [14] are among the most popular formalisms to express properties of systems both in the formal verification and artificial intelligence communities.  $LTL_f$  has also recently gained traction in business process modeling (BPM) [31, 22], where the real execution of a (business) process is assumed to be always finite.

Due to its propositional nature, LTL is inherently limited to the modeling of finite-state systems, while many real-world scenarios, e.g., systems involving numeric data or data-aware processes [5, 6, 8], are better modeled as infinite-state systems, for which a *first-order* setting is needed. Thus, various first-order extensions of LTL have been studied in the literature. Generally speaking, existing results in this direction are either purely theoretical (e.g. [32]), or they have been developed with specific practical scenarios in mind and appear difficult to apply to more general ones (e.g. [12, 13, 19]).

As a coherent and principled approach to mitigate this situation, the logic of  $LTL_f$  modulo theories ( $LTL_f^{MT}$ ) has been recently introduced [24].  $LTL_f^{MT}$  extends  $LTL_f$  by replacing propositions with general first-order formulas interpreted over arbitrary theories, similar to how *satisfiability modulo theories* (SMT) extends the Boolean satisfiability problem, and by allowing comparisons between first-order variables referring to possibly different time points.

In general,  $LTL_f^{MT}$  is undecidable, and it has been shown to be semi-decidable if applied to decidable first-order fragments and/or theories [24, 25]: Crucially, the semi-decidability result has been shown by providing an effective SMT-based encoding of a tree-

shaped tableau that, once implemented in the BLACK temporal reasoning system [27, 28], has proved to work well in practice. Moreover, being theory-agnostic, the technique works in many different scenarios, leveraging the many expressive theories, and combinations thereof, supported by modern SMT solvers [2]. Hence,  $LTL_f^{MT}$  provides a general and theoretically well-founded common ground for first-order temporal logics that, at the same time, can be applied to complex scenarios. The *satisfiability* problem asks whether for a given temporal logic formula  $\phi$  there exists a trace that satisfies  $\phi$ . Satisfiability is a central problem in linear-time temporal logics since a range of key verification tasks, including model checking, can be reduced to it [38, 34].

While undecidability is unavoidable when considering expressive infinite-state systems and logics to describe them [4, 3, 30, 18], reasoning and verification has been shown decidable in several specific cases [6, 22, 17, 13]. It is thus natural to ask which fragments of  $LTL_f^{MT}$  have a decidable satisfiability problem.

In this paper, we address this question in a general way. First, we extend the tree-shaped tableau for  $LTL_f^{MT}$  provided in [24] with a *pruning rule* that guarantees soundness and *completeness* for any *decidable* first-order theory, and we give a very general semantic, sufficient condition, called *finite memory*, that guarantees that the tableau, augmented with the new rule, is finite (hence, that its construction terminates). This equips  $LTL_f^{MT}$  with a sound and complete semi-decision procedure that, in particular, is guaranteed to terminate for any formula that satisfies the finite memory property.

In the next step, we identify a number of syntactic fragments of  $LTL_f^{MT}$  that satisfy the finite memory property, and are therefore decidable. In this way, we both derive novel decidability results, and recast and generalise existing ones in this framework. In particular, we prove decidability for  $LTL_f^{MT}$  formulas that either: do not compare variables at different time points; only use temporal operators F, X, and  $\bar{X}$ ; belong to a *bounded lookback* fragment that restrict variable dependencies in a way to require only a bounded amount of memory; or that are interpreted over arithmetic theories but with first-order subformulas restricted to variable-to-variable/constant comparisons.

A crucial feature of the new pruning rule is that it is sound and complete in the general case. It is hence always applicable, avoiding the need to identify the fragment of the input formula beforehand. This feature will ease implementation (which we leave for future work), because a single procedure can be implemented, and optimized, that works for a wide range of decidable fragments as well as for the semi-decidable general case. These results further improve

the applicability of  $\text{LTL}_f^{\text{MT}}$  in many scenarios involving complex infinite-state systems, e.g. verification tasks from the areas of knowledge representation or BPM [6, 5, 22, 13, 19]. Moreover, one may lift the known connection between automated planning and propositional LTL [1, 9] to a first-order, data-aware setting, and use  $\text{LTL}_f^{\text{MT}}$  to address planning problems based on expressive theories.

The paper is structured as follows. We introduce the relevant background in Section 2. Then, Section 3 provides the new pruning rule for  $\text{LTL}_f^{\text{MT}}$  and proves that it maintains soundness and completeness. Section 4 defines the condition of *finite memory*, proves the termination of the tableau for formulas satisfying such condition, and identifies a number of decidable fragments of  $\text{LTL}_f^{\text{MT}}$ . Finally, Section 5 concludes discussing related work and future directions. For reasons of space, some proofs were moved to an extended version [26].

## 2 Background

We consider a given first-order multi-sorted *signature*  $\Sigma = \langle \mathcal{S}, \mathcal{P}, \mathcal{F}, \mathcal{V}, \mathcal{W} \rangle$ , where  $\mathcal{S}$  is a set of sorts;  $\mathcal{P}$  is a set of predicate and  $\mathcal{F}$  a set of function symbols;  $\mathcal{V}$  is a finite, non-empty set of *data variables*; and  $\mathcal{W}$  is a set of variables disjoint from  $\mathcal{V}$  that will be used for quantification; all variables are associated with a sort in  $\mathcal{S}$ . Each predicate and function symbol is supposed to have a type taking sorts from  $\mathcal{S}$ ; constant symbols are represented by zero-ary function symbols. We assume that  $\Sigma$  contains equality predicates for all sorts.

Then,  $\Sigma$ -terms  $t$  are built according to the following grammar:

$$t := v \mid w \mid f(t_1, \dots, t_k) \mid \bigcirc v \mid \bigodot v$$

where  $v \in \mathcal{V}$ ,  $w \in \mathcal{W}$ ,  $f \in \mathcal{F}$  has arity  $k$ , and each  $t_i$  is a term of appropriate sort. Intuitively,  $\bigcirc$  and  $\bigodot$  are the *next* and *weak next* operators, that represent the value of a variable  $v \in \mathcal{V}$  in the next state (see the semantics below). An atom is of the form  $p(t_1, \dots, t_k)$ , where  $p \in \mathcal{P}$  is a predicate symbol of arity  $k$ , and  $t_i$  are terms of appropriate sort. Then,  $\text{LTL}_f^{\text{MT}}$  formulas are defined as follows:

$$\begin{aligned} \lambda &:= a \mid \neg a \mid \lambda_1 \wedge \lambda_2 \mid \lambda_1 \vee \lambda_2 \mid \exists w. \lambda \mid \forall w. \lambda \\ \phi &:= \top \mid \lambda \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \mathbf{X}\phi \mid \tilde{\mathbf{X}}\phi \mid \phi_1 \mathbf{U} \phi_2 \mid \phi_1 \mathbf{R} \phi_2 \end{aligned}$$

where  $a$  is an atom and  $w \in \mathcal{W}$ . Formulas  $\lambda$  as above are called *first-order formulas*. We call  $\phi$  a *state formula* if all its free variables are in  $\mathcal{V}$ .  $\Sigma$ -formulas without free variables are  $\Sigma$ -*sentences*, and a set of  $\Sigma$ -sentences is a  $\Sigma$ -*theory*  $\mathcal{T}$ . Note the difference between the *next* ( $\bigcirc$ ) and *weak next* ( $\bigodot$ ) operators, acting on variables, and the *tomorrow* ( $\mathbf{X}$ ), and *weak tomorrow* ( $\tilde{\mathbf{X}}$ ) temporal operators, acting on formulas.

To define the semantics of first-order formulas, we use the standard notion of a  $\Sigma$ -*structure*  $M$ , which associates each sort  $s \in \mathcal{S}$  with a domain  $s^M$ , and each predicate  $p \in \mathcal{P}$  and function symbol  $f \in \mathcal{F}$  with a suitable interpretation  $p^M$  and  $f^M$ . The equality predicates have the usual interpretation given by the identity relation. The carrier of  $M$ , i.e., the union of all domains of sorts in  $\mathcal{S}$ , is denoted by  $|M|$ . A function  $\alpha: \mathcal{V} \rightarrow |M|$  is a *state variable assignment* with respect to  $M$ , while a function  $\gamma: \mathcal{W} \rightarrow |M|$  is an *environment*, where we assume in both cases that all variables are mapped to elements of their domain. We write  $\gamma[u \mapsto e]$  for the environment  $\gamma$  extended with a binding from  $u$  to  $e$ . A *run* is a pair  $\sigma = (M, \langle \alpha_0, \dots, \alpha_{n-1} \rangle)$  of a  $\Sigma$ -structure  $M$  and a sequence of state variable assignments with respect to  $M$ , and  $|\sigma| = n$  is its length.

**Example 1.** Let  $\mathcal{V}$  consist of variables  $x$  and  $y$  of sort  $\text{int}$ , and  $M$  be the (unique) model of the theory of linear arithmetic over the integers (LIA). Then, e.g.,  $(M, \bar{\alpha})$  is a run of length 3, for  $\bar{\alpha} = \{x \mapsto -1, y \mapsto 0\}, \{x \mapsto 0, y \mapsto 1\}, \{x \mapsto 2, y \mapsto 2\}$ .

For such a run  $\sigma$ , some  $i$  with  $0 \leq i < n$ , and an environment  $\gamma$ , a term  $t$  is *well-defined* if  $i < n-1$ , or  $t$  does not contain subterms of the form  $\bigcirc v$  or  $\bigodot v$ . In this case, the evaluation of the term  $t$  is denoted  $\llbracket t \rrbracket_{\sigma, \gamma}^i$ , and defined as follows:

$$\begin{aligned} \llbracket v \rrbracket_{\sigma, \gamma}^i &= \alpha_i(v) & \llbracket \bigcirc v \rrbracket_{\sigma, \gamma}^i &= \llbracket \bigodot v \rrbracket_{\sigma, \gamma}^i = \alpha_{i+1}(v) \\ \llbracket w \rrbracket_{\sigma, \gamma}^i &= \gamma(w) & \llbracket f(t_1, \dots, t_k) \rrbracket_{\sigma, \gamma}^i &= f^M(\llbracket t_1 \rrbracket_{\sigma, \gamma}^i, \dots, \llbracket t_k \rrbracket_{\sigma, \gamma}^i) \end{aligned}$$

where  $v \in \mathcal{V}$  and  $w \in \mathcal{W}$ . Satisfaction of a first-order formula  $\lambda$  with respect to an environment  $\gamma$  in the run  $\sigma$  with  $i < |\sigma|$ , denoted  $\sigma \models_{\gamma}^i \lambda$ , is defined as follows:

$$\begin{aligned} \sigma \models_{\gamma}^i p(t_1, \dots, t_k) & \quad \text{if } t_1, \dots, t_k \text{ are well-defined and} \\ & \quad (\llbracket t_1 \rrbracket_{\sigma, \gamma}^i, \dots, \llbracket t_k \rrbracket_{\sigma, \gamma}^i) \in p^M, \text{ or} \\ & \quad \text{if some } t_1, \dots, t_k \text{ is not well-defined and} \\ & \quad t_1, \dots, t_k \text{ contain } \bigodot \text{ but do not contain } \bigcirc \\ \sigma \models_{\gamma}^i \neg p(t_1, \dots, t_k) & \quad \text{if } \sigma \not\models_{\gamma}^i p(t_1, \dots, t_k) \\ \sigma \models_{\gamma}^i \lambda_1 \wedge \lambda_2 & \quad \text{if } \sigma \models_{\gamma}^i \lambda_1 \text{ and } \sigma \models_{\gamma}^i \lambda_2 \\ \sigma \models_{\gamma}^i \lambda_1 \vee \lambda_2 & \quad \text{if } \sigma \models_{\gamma}^i \lambda_1 \text{ or } \sigma \models_{\gamma}^i \lambda_2 \\ \sigma \models_{\gamma}^i \exists w. \lambda & \quad \text{if } \sigma \models_{\gamma[w \mapsto e]}^i \lambda \text{ for some } e \in s^M \\ \sigma \models_{\gamma}^i \forall w. \lambda & \quad \text{if } \sigma \models_{\gamma[w \mapsto e]}^i \lambda \text{ for all } e \in s^M \end{aligned}$$

where  $w$  is assumed to have sort  $s$ . Satisfaction with respect to  $\sigma$  is extended to a general  $\text{LTL}_f^{\text{MT}}$  formula  $\phi$  as follows:

$$\begin{aligned} \sigma \models^i \lambda & \quad \text{if } \sigma \models_{\emptyset}^i \lambda \\ \sigma \models^i \phi_1 \wedge \phi_2 & \quad \text{if } \sigma \models^i \phi_1 \text{ and } \sigma \models^i \phi_2 \\ \sigma \models^i \phi_1 \vee \phi_2 & \quad \text{if } \sigma \models^i \phi_1 \text{ or } \sigma \models^i \phi_2 \\ \sigma \models^i \mathbf{X}\phi & \quad \text{if } i < |\sigma| - 1 \text{ and } \sigma \models^{i+1} \phi \\ \sigma \models^i \tilde{\mathbf{X}}\phi & \quad \text{if } i = |\sigma| - 1 \text{ or } \sigma \models^{i+1} \phi \\ \sigma \models^i \phi_1 \mathbf{U} \phi_2 & \quad \text{if there is some } j, i \leq j < |\sigma| \text{ such that } \sigma \models^j \phi_2 \\ & \quad \text{and } \sigma \models^k \phi_1 \text{ for all } i \leq k < j \\ \sigma \models^i \phi_1 \mathbf{R} \phi_2 & \quad \text{if either } \sigma \models^j \phi_2 \text{ for all } i \leq j < |\sigma|, \text{ or there is} \\ & \quad \text{some } j, i \leq j < |\sigma| \text{ such that } \sigma \models^j \phi_1 \\ & \quad \text{and } \sigma \models^k \phi_2 \text{ for all } i \leq k \leq j \end{aligned}$$

Finally,  $\sigma$  *satisfies*  $\phi$ , denoted by  $\sigma \models \phi$ , if  $\sigma \models^0 \phi$  holds. We use the usual shorthands  $\mathbf{F}\phi \equiv (\top \mathbf{U} \phi)$  and  $\mathbf{G}\phi \equiv (\perp \mathbf{R} \phi)$ , where  $\top \equiv a \vee \neg a$  for any atom  $a$  and  $\perp \equiv \neg \top$ . For instance, the run in Ex. 1 satisfies  $(y \geq x) \mathbf{U} (x = y)$  and  $\mathbf{G}(\bigodot x > x)$ , but not  $\mathbf{G}(\bigcirc x > x)$  as no first-order formula with  $\bigcirc$  holds in the last instant.

Let  $\mathcal{V}^{\bigcirc} = \{\bigcirc v \mid v \in \mathcal{V}\}$  be the set of all the *next* variables of  $\mathcal{V}$ , and similarly for  $\mathcal{V}^{\bigodot}$ . A first-order formula  $\phi$  without  $\mathcal{V}^{\bigcirc} \cup \mathcal{V}^{\bigodot}$  is satisfied by some  $\Sigma$ -structure  $M$  and state variable assignment  $\alpha: \mathcal{V} \rightarrow |M|$ , denoted  $M, \alpha \models \phi$ , if  $(M, \langle \alpha \rangle) \models \phi$ , which corresponds to the usual notion of first-order satisfaction; if  $\phi$  is a sentence, we simply write  $M \models \phi$ . For a  $\Sigma$ -structure  $M$ , we will write  $M \in \mathcal{T}$  to express that  $M$  is a model of  $\mathcal{T}$ . A formula is called  $\mathcal{T}$ -satisfiable if it is satisfied by some  $\sigma = (M, \bar{\alpha})$  with  $M \in \mathcal{T}$ . Moreover, two first-order formulas  $\phi_1$  and  $\phi_2$  are  $\mathcal{T}$ -*equivalent*, denoted  $\phi_1 \equiv_{\mathcal{T}} \phi_2$ , if  $\neg(\phi_1 \leftrightarrow \phi_2)$  is not  $\mathcal{T}$ -satisfiable.

A  $\Sigma$ -theory  $\mathcal{T}$  has *quantifier elimination* (QE) if for any  $\Sigma$ -formula  $\phi$  there is a quantifier-free formula  $\phi'$  that is  $\mathcal{T}$ -equivalent to  $\phi$ .

In the paper we will sometimes refer to common SMT theories [2]: the theory of equality and uninterpreted functions for a given  $\Sigma$  (EUF), linear arithmetics over rationals (LRA) and integers (LIA).

**Tableau for  $\text{LTL}_f^{\text{MT}}$**  We now recall the one-pass tree-shaped tableau for  $\text{LTL}_f^{\text{MT}}$  presented in [24]. The closure of a formula  $\phi$ ,

denoted  $\mathcal{C}(\phi)$ , is the smallest set of formulas that contains all sub-formulas of  $\phi$ , and, in addition,  $\mathsf{X}(\phi_1 \mathsf{U} \phi_2)$  whenever  $\phi_1 \mathsf{U} \phi_2 \in \mathcal{C}(\phi)$  and  $\mathsf{X}(\psi_1 \mathsf{R} \psi_2)$  whenever  $\psi_1 \mathsf{R} \psi_2 \in \mathcal{C}(\phi)$ . A *tableau* for an  $\text{LTL}_f^{\text{MT}}$  formula  $\phi$  is a rooted tree in which each node  $u$  is labelled by a set of formulas  $\Gamma(u) \subseteq \mathcal{C}(\phi)$ , as follows. The root node  $u_0$  has label  $\Gamma(u_0) = \{\phi\}$ , and every other node is the result of applying one of a set of rules to its parent. If any is applicable, one of the *expansion rules*, shown in Tab. 1, is applied.

rule	$\phi \in \Gamma(u)$	$\Gamma_1(\phi)$	$\Gamma_2(\phi)$
DISJUNCTION	$\psi \vee \chi$	$\{\psi\}$	$\{\chi\}$
CONJUNCTION	$\psi \wedge \chi$	$\{\psi, \chi\}$	
UNTIL	$\psi \mathsf{U} \chi$	$\{\chi\}$	$\{\psi, \mathsf{X}(\psi \mathsf{U} \chi)\}$
RELEASE	$\psi \mathsf{R} \chi$	$\{\psi, \chi\}$	$\{\chi, \mathsf{X}(\psi \mathsf{R} \chi)\}$

**Table 1.** Expansion rules for  $\text{LTL}_f^{\text{MT}}$  tableau.

When applying a rule to a formula  $\phi \in \Gamma(u)$  for a node  $u$ , two children  $u_1$  and  $u_2$  of  $u$  are constructed, which are labeled  $\Gamma(u) \setminus \{\phi\} \cup \Gamma_1(\phi)$  and  $\Gamma(u) \setminus \{\phi\} \cup \Gamma_2(\phi)$ , respectively, with the second child omitted if  $\Gamma_2(\phi)$  is empty. If no expansion rule is applicable to a node  $u$ , the node is called *poised*. By definition of the expansion rules, such a node can contain only atoms, or temporal formulas rooted by  $\mathsf{X}$  and  $\tilde{\mathsf{X}}$ . Poised nodes represent a state in a possible model for the formula. Then, *time* advances, from a poised node  $u$ , by applying the STEP rule, which creates a child  $u'$  of  $u$  such that:

$$\text{STEP: } \Gamma(u') = \{\psi \mid \mathsf{X}\psi \in \Gamma(u) \text{ or } \tilde{\mathsf{X}}\psi \in \Gamma(u)\}$$

However, the STEP rule is only applied if the branch is not ready to be either *accepted* or *rejected* by one of two *termination rules*. These rules are defined, for a branch  $\bar{u}$ , via a first-order formula  $\Omega(\bar{u})$  which summarizes all constraints along the branch. The formula is defined over the signature  $\Sigma' = (\mathcal{S}, \mathcal{P}', \mathcal{F}, \mathcal{V}^\omega, \mathcal{W})$ , where  $\mathcal{P}' = \mathcal{P} \cup \{\ell\}$  for some fresh  $\ell$ , and  $\mathcal{V}^\omega = \bigcup_{i \in \mathbb{N}} V^i$  where  $V^i = \{v^i \mid v \in \mathcal{V}\}$  are indexed versions of the variables in  $\mathcal{V}$ . We write  $\bar{\mathcal{V}}$  for the list of variables  $(v_1, \dots, v_k)$ , ordering the variables in  $\mathcal{V}$  in some arbitrary but fixed way; and similarly,  $\bar{V}^i$  for  $(v_1^i, \dots, v_k^i)$ .

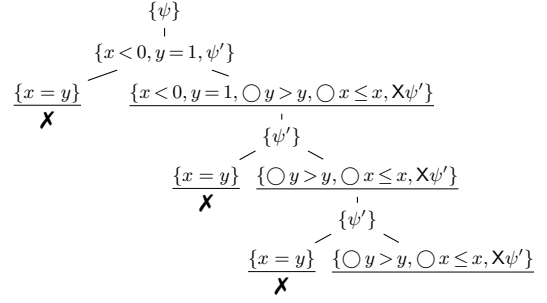
The stepped version  $t^{(i)}$  of an arbitrary term  $t$  is defined as follows: 1.  $w^{(i)} = w$  for all  $w \in \mathcal{W}$ ; 2.  $v^{(i)} = v^i$  for all  $v \in \mathcal{V}$ ; 3.  $(\bigcirc x)^{(i)} = (\bigcirc x)^{(i)}$ ; 4.  $f(t_1, \dots, t_n)^{(i)} = f(t_1^{(i)}, \dots, t_n^{(i)})$ . We extend the notion to formulas, and set  $\psi^{(i)}$  to the formula obtained from  $\psi$  by replacing each term  $t$  in  $\psi$  by  $t^{(i)}$ . The role of  $\ell$  is to denote the last position of a run. Given a first-order formula  $\phi$ , the formula  $L(\phi)$  is obtained by replacing all atoms  $A$  containing any term from  $\mathcal{V}^\bigcirc$  by  $\ell \wedge A$ , and all atoms  $B$  containing any term from  $\mathcal{V}^\ominus$  (but not from  $\mathcal{V}^\bigcirc$ ) by  $\ell \rightarrow B$ .

More generally, we define  $\Omega$  for sequences of constraints. Let  $\bar{C} = \langle C_0, \dots, C_{m-1} \rangle$  be a sequence of first-order formulas with free variables  $\mathcal{V} \cup \mathcal{V}^\bigcirc \cup \mathcal{V}^\ominus$ . Then  $\Omega(\bar{C})$  is defined as

$$\Omega(\bar{C}) = \bigwedge_{i=0}^{m-2} C_i^{(i)} \wedge L(C_{m-1})^{(m-1)}$$

Notice that, according to the definition of  $\psi^{(i)}$ , only variables from  $\mathcal{V}$  are stepped, whereas the  $\ell$  atom is left unchanged. For a branch  $\bar{u}$  with poised nodes  $\bar{\pi} = \langle \pi_0, \dots, \pi_{m-1} \rangle$  and  $F(\pi_i)$  the conjunction of first-order formulas in  $\pi_i$ , we set  $\Omega(\bar{u}) = \Omega(\langle F(\pi_0), \dots, F(\pi_{m-1}) \rangle)$ .<sup>1</sup> Intuitively,  $\Omega(\bar{u})$  serves the purpose to capture a candidate model along the branch  $\bar{u}$ .

<sup>1</sup> We use a slightly modified but equivalent variant of the definition of  $\Omega$  from [24], applying the  $L$  operator only to the last instant. This allows us to use a single constant  $\ell$ , which will simplify the definition of the PRUNE rule.



**Figure 1.** Example tableau for the  $\text{LTL}_f^{\text{MT}}$  formula  $\psi$  of Ex. 2. Poised nodes are underlined, and nodes marked  $\mathsf{X}$  are rejected.

Given  $\Omega(\bar{u})$ , the termination rules are defined as follows. The EMPTY rule is responsible for acceptance:

EMPTY: If  $\Gamma(\pi_{m-1})$  does not contain formulas rooted by  $\mathsf{X}$  and  $\Omega(\bar{u}) \wedge \neg \ell$  is satisfiable, the branch is accepted.

Whereas the CONTRADICTION rule is responsible for rejection:

CONTRADICTION: If  $\Omega(\bar{u})$  is  $\mathcal{T} \cup \text{EUF}$ -unsatisfiable, the branch  $\bar{u}$  is rejected.

From [24], we can state the soundness and completeness of the tableau for  $\text{LTL}_f^{\text{MT}}$  so defined.

**Proposition 1** ([24]). *A tableau for an  $\text{LTL}_f^{\text{MT}}$  formula  $\phi$  contains an accepted branch if and only if the formula is satisfiable.*

The construction of the tableau for an arbitrary formula is not guaranteed to terminate, which is to be expected since  $\text{LTL}_f^{\text{MT}}$  is in general undecidable. However, since accepted branches are finite, if the formula is satisfiable, a breadth-first construction of the tree will surely find an accepted branch. Hence, for decidable theories, this tableau provides a *semi-decision* procedure for  $\text{LTL}_f^{\text{MT}}$  satisfiability.

**Proposition 2** ([24]).  *$\text{LTL}_f^{\text{MT}}$  satisfiability is semi-decidable.*

**Example 2.** Consider the following formula:

$$\psi := (x < 0 \wedge y = 1) \wedge ((\bigcirc y > y \wedge \bigcirc x \leq x) \mathsf{U} x = y)$$

interpreted over LRA. A partial tableau for  $\psi$  is shown in Fig. 1, where  $\psi' := (\bigcirc y > y \wedge \bigcirc x \leq x) \mathsf{U} x = y$ . Note that  $\psi$  is unsatisfiable, but the CONTRADICTION rule is not sufficient to conclude that, as the right-most branch is going to expand forever.

### 3 A new pruning rule for the $\text{LTL}_f^{\text{MT}}$ tableau

As discussed in Ex. 2, the right-most branch of Fig. 1 is the prototypical example of a branch that expands forever because of some unfulfillable request that is postponed forever without ever causing a local contradiction. In Reynolds' tree-shaped tableau for propositional LTL, this case is handled by an ad-hoc PRUNE rule, which takes care of rejecting such branches [37]. Here, we define a similar rule for  $\text{LTL}_f^{\text{MT}}$ . To this end, we use a quantified variant of the formula  $\Omega(\bar{C})$ , for a sequence of first-order formulas  $\bar{C}$ .

**Definition 1** (History constraints). *The history constraint of a sequence of first-order formulas  $\bar{C}$ , denoted  $h(\bar{C})$ , is defined as:*

$$h(\bar{C}) = \begin{cases} \top & \text{if } \bar{C} \text{ is empty} \\ (\exists V^0 \dots V^{m-1}. \Omega(\bar{C}))[\bar{V}^m / \bar{\mathcal{V}}] & \text{otherwise} \end{cases}$$

That is, all stepped variables are existentially quantified except for the last ones, which are renamed to  $\bar{\mathcal{V}}$ , so that  $h(\bar{C})$  is a formula with free variables  $\mathcal{V}$ . For a branch  $\bar{u}$  with poised nodes  $\bar{\pi} = \langle \pi_0, \dots, \pi_{m-1} \rangle$ , let  $h(\bar{\pi}) = h(\langle F(\pi_0), \dots, F(\pi_{m-1}) \rangle)$ . Intuitively, the history constraint of a branch  $\bar{u}$  summarises all constraints accumulated along the branch, just like  $\Omega$ , but by existentially quantifying all variables except those in the last instant, it expresses the *effect* of the accumulated constraints (the history) on the variables  $\mathcal{V}$ . If the theory under consideration has quantifier elimination (QE), history constraints are always equivalent to quantifier-free formulas.

**Example 3.** Let  $\langle \pi_0, \pi_1, \pi_2 \rangle$  be the poised nodes in the right-most branch of the tableau in Fig. 1, and denote as  $\bar{\pi}_{\leq i}$ , for  $0 \leq i \leq 2$ , the branches up to these nodes. Then, we have:

$$\begin{aligned} h(\bar{\pi}_{\leq 0}) &= (\exists x_0 y_0. x_0 < 0 \wedge y_0 = 1 \wedge y > y_0 \wedge x \leq x_0 \wedge \ell) \\ &\equiv_{\text{LRA}} x < 0 \wedge y > 1 \wedge \ell \\ h(\bar{\pi}_{\leq 1}) &= \exists x_1 y_1 x_0 y_0. x_0 < 0 \wedge y_0 = 1 \wedge y_1 > y_0 \wedge x_1 \leq x_0 \wedge \\ &\quad y > y_1 \wedge x \leq x_1 \wedge \ell \\ &\equiv_{\text{LRA}} x < 0 \wedge y > 1 \wedge \ell \\ h(\bar{\pi}_{\leq 2}) &\equiv_{\text{LRA}} \exists x_2 y_2. x_2 < 0 \wedge y_2 > 1 \wedge y > y_2 \wedge x \leq x_2 \wedge \ell \\ &\equiv_{\text{LRA}} x < 0 \wedge y > 1 \wedge \ell \end{aligned}$$

Here the equivalences are obtained with quantifier elimination in LRA, so all history constraints are LRA-equivalent. This reflects the fact that what can be said about  $x$  and  $y$  after the respective nodes is always the same:  $x$  is negative, and  $y$  is greater than 1.

Intuitively, if the labels and history constraints of nodes repeat, no progress is made on this branch. This motivates the next definition.

Given a tableau branch  $\bar{u}$  with poised nodes  $\bar{\pi} = \langle \pi_0, \dots, \pi_{m-1} \rangle$ :

**PRUNE:** If  $\Gamma(\pi_i) = \Gamma(\pi_{m-1})$  for some  $i < m$  and  $h(\bar{\pi}) \models_{\mathcal{T}} h(\bar{\pi}_{\leq i})$  then  $\bar{u}$  is rejected.

Testing whether the PRUNE rule applies requires to check entailment in the underlying theory  $\mathcal{T}$ . If  $\mathcal{T}$  is decidable, this is always possible (e.g., if  $\mathcal{T}$  is LIA or LRA). However, in Sec. 4 we show that even for theories where this is not feasible in general, PRUNE can be applied in a number of special cases. Moreover, note that the entailment condition of the PRUNE rule is equivalent to saying that the set of states described by the formula  $h(\bar{\pi})$  (which represents the history effect at the end of  $\bar{\pi}$ ) is contained in the set of states described by the formula  $h(\bar{\pi}_{\leq i})$  (representing the effect up to instant  $i$ ).

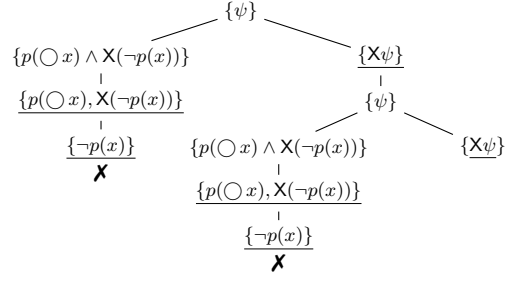
Finally, note that even though there is an apparent overlap between the definitions of the EMPTY and PRUNE rules, the two can never be applicable together on the same node, because in this case, EMPTY would have triggered before (on the repeated node identified by PRUNE), and the branch would have been already accepted.

The rightmost branch in Fig. 1 is rejected by the PRUNE rule: for  $\pi_1$  and  $\pi_2$  the last two poised nodes on the branch,  $\Gamma(\pi_1) = \Gamma(\pi_2)$  holds and, as shown in Ex. 3,  $h(\bar{\pi}_{\leq 1})$  and  $h(\bar{\pi}_{\leq 2})$  are LRA-equivalent. A further example of an application of the rule follows.

**Example 4.** Consider the following unsatisfiable formula interpreted over EUF, for a unary predicate  $p$ :

$$\psi := F(p(\bigcirc x) \wedge X(\neg p(x)))$$

The corresponding tableau is shown in Fig. 2. Let  $\bar{u}$  be the right-most branch with poised nodes  $\bar{\pi} = \langle \pi_0, \pi_1 \rangle$ . We have  $\Gamma(\pi_0) = \Gamma(\pi_1)$ , and  $h(\bar{\pi}_{\leq 0}) = h(\bar{\pi}_{\leq 1}) = \top$ . Thus the PRUNE rule applies, and  $\bar{u}$  can be rejected.



**Figure 2.** Example of application of the PRUNE rule from Ex. 4.

Since the PRUNE rule can only reject (but not accept) branches, it may only affect *completeness*, but not *soundness*. As we prove in the remainder of this section, completeness of the tableau calculus of [24] is indeed preserved when augmented with the PRUNE rule.

**Completeness** Here, we extend the completeness result of [24, 25] to account for the additional PRUNE rule. We start by defining a *pre-model*, an abstract structure summarising the important aspects of a state sequence in a tableau branch.

**Definition 2 (Atom).** An atom  $\Delta$  for an  $\text{LTL}_f^{\text{MT}}$  formula  $\phi$  is a set  $\Delta \subseteq \mathcal{C}(\phi)$  such that:

1. the conjunction of all first-order formulas in  $\Delta$  is  $\mathcal{T}$ -satisfiable;
2. for all  $\psi \in \Delta$  to which a rule from Tab. 1 applies, either  $\Gamma_1 \subseteq \Delta$ , or  $\Gamma_2 \neq \emptyset$  and  $\Gamma_2 \subseteq \Delta$ ; and
3.  $\Delta$  is closed under logical deduction as far as  $\mathcal{C}(\phi)$  is concerned.

**Definition 3.** A pre-model for  $\phi$  is a sequence of atoms  $\bar{\Delta} = \langle \Delta_0, \dots, \Delta_{n-1} \rangle$  such that  $\phi \in \Delta_0$ , and for all  $i$ ,  $0 \leq i < n$ :

1.  $\Delta_{n-1}$  does not contain any  $p(t_1, \dots, t_k)$  where  $\mathcal{V}^\circ$  occurs,
2. if  $X\phi' \in \Delta_i$  then  $i < n - 1$  and  $\phi' \in \Delta_{i+1}$ ,
3. if  $\bar{X}\phi' \in \Delta_i$  then  $i = n - 1$  or  $\phi' \in \Delta_{i+1}$ ,
4. if  $\phi_1 \cup \phi_2 \in \Delta_i$  then there is some  $i \leq j < n$  such that  $\phi_2 \in \Delta_j$  and  $\phi_1 \in \Delta_k$  for all  $i \leq k < j$ ,
5. if  $\phi_1 \text{ R } \phi_2 \in \Delta_i$  then either  $\phi_2 \in \Delta_k$  for all  $i \leq k < n$ , or there is some  $i \leq j < n$  such that  $\phi_1 \in \Delta_j$  and  $\phi_2 \in \Delta_k$  for all  $i \leq k \leq j$ , and
6. all  $\Delta_i$  are minimal with respect to set inclusion.

Let  $F(\Delta)$  be the conjunction of all first-order formulas in an atom  $\Delta$ . Given a pre-model  $\bar{\Delta} = \langle \Delta_0, \dots, \Delta_{n-1} \rangle$ , we say that  $\bar{\Delta}$  is *satisfiable* if  $\Omega(\langle F(\Delta_0), \dots, F(\Delta_{n-1}) \rangle) \wedge \neg \ell$  is  $\mathcal{T}$ -satisfiable.

Following [24, 25], one can show that from any pre-model for an  $\text{LTL}_f^{\text{MT}}$  formula  $\phi$  one can obtain a model of  $\phi$ , and *vice versa*, any model of  $\phi$  can be represented by a pre-model:

**Proposition 3** ([24, 25]). An  $\text{LTL}_f^{\text{MT}}$  formula  $\phi$  is satisfiable if and only if it has a satisfiable pre-model.

There is a precise connection between pre-models of a formula and branches of the tableau. In particular, the following extraction lemma can be proved, as in [25, Lem. 2 in Appendix A].

For a node  $u$  in a tableau for  $\phi$ , let the *atom* of  $u$ , denoted  $\Delta(u)$ , be the set of all formulas in  $\mathcal{C}(\phi)$  that are entailed by  $\Gamma(u)$ .

**Proposition 4** ([24, 25]). If  $\bar{\Delta} = \langle \Delta_0, \dots, \Delta_{n-1} \rangle$  is a satisfiable pre-model for  $\phi$ , every complete tableau for  $\phi$  has a branch with step nodes  $\bar{\pi} = \langle \pi_0, \dots, \pi_{n-1} \rangle$  such that  $\Delta(\pi_i) = \Delta_i$  for all  $0 \leq i < n$ .

To prove completeness, we have to show that if a formula  $\phi$  is satisfiable, there is an accepted branch. As  $\phi$  is satisfiable, it

has a model, and by Prop. 3, there is also a satisfiable pre-model  $\bar{\Delta} = \langle \Delta_0, \dots, \Delta_{n-1} \rangle$  for  $\phi$ . Thus, by Prop. 4, there is a branch  $\bar{\pi} = \langle \pi_0, \dots, \pi_{n-1} \rangle$  in the tableau such that  $\Delta(\pi_i) = \Delta_i$  for all  $i$ ,  $0 \leq i < n$ . It is easy to see that (a prefix of)  $\bar{\pi}$  cannot be rejected by the CONTRADICTION rule, as otherwise  $\bar{\Delta}$  would not be a satisfiable pre-model. However, it remains to show that  $\bar{\pi}$  cannot be rejected by the PRUNE rule. To this end, we first define a *redundant segment* of a pre-model, i.e., a segment that can be safely removed from a satisfiable pre-model to obtain another, shorter, satisfiable pre-model. Then, we show that if there are no redundant segments, the tableau branch extracted by Prop. 4 cannot be rejected by PRUNE. To do so, we extend our notion of history constraints to pre-models in a natural way, that is, given a pre-model  $\bar{\Delta} = \langle \Delta_0, \dots, \Delta_{n-1} \rangle$ , we define  $h(\bar{\Delta}) = h(\langle F(\Delta_0), \dots, F(\Delta_{n-1}) \rangle)$ .

**Definition 4** (Redundant segment). *Let  $\Delta = \langle \Delta_0, \dots, \Delta_{n-1} \rangle$  be a pre-model for  $\psi$  and  $j < k < n$ . Then the subsequence  $\bar{\Delta}_{[j+1, k]}$  is redundant if  $\Delta_j = \Delta_k$  and  $h(\bar{\Delta}_{\leq k}) \models_{\mathcal{T}} h(\bar{\Delta}_{\leq j})$ .*

Intuitively, a redundant segment can be removed from a pre-model because it does no useful work towards the satisfaction of the formula. To show this, we need an auxiliary result about history constraints. First, given two state variable assignments  $\alpha$  and  $\alpha'$  we define the combination  $\alpha \otimes \alpha'$  of them as a variable assignment with domain  $V \cup \mathcal{V}^{\circ} \cup \mathcal{V}^{\ominus}$  by setting  $(\alpha \otimes \alpha')(v) = \alpha(v)$  and  $(\alpha \otimes \alpha')(\circ v) = (\alpha \otimes \alpha')(\ominus v) = \alpha'(v)$  for all  $v \in V$ . That is,  $\alpha$  is used to interpret the current state variables, and  $\alpha'$  to interpret the variables at the next state. Let  $\bar{C} = \langle C_0, \dots, C_{m-1} \rangle$  be a sequence of first-order formulas with free variables  $V \cup \mathcal{V}^{\circ} \cup \mathcal{V}^{\ominus}$ . Given a model  $M$ , and a sequence of state variable assignments  $\bar{\alpha} = \langle \alpha_0, \dots, \alpha_m \rangle$ , we write  $M, \bar{\alpha} \models \bar{C}$  if  $M, \alpha_i \otimes \alpha_{i+1} \models C_i$  for all  $0 \leq i < m-1$ , and  $M, \alpha_{m-1} \otimes \alpha_m \models L(C_{m-1})$ . We then have the following relationship between satisfying assignments for history constraints, and sequences of assignments that satisfy each constraint in the sequence individually (similar as [22, Lemma 3.5]):

**Lemma 1.** *Let  $M$  be a  $\Sigma$ -structure and  $\bar{C} = \langle C_0, \dots, C_{m-1} \rangle$  be a sequence of first-order formulas with free variables  $\mathcal{V} \cup \mathcal{V}^{\circ} \cup \mathcal{V}^{\ominus}$ , for  $m \geq 1$ .*

- (1) *If  $M, \langle \alpha_0, \dots, \alpha_m \rangle \models \bar{C}$  then  $M, \alpha_m \models h(\bar{C})$ .*
- (2) *If  $M, \alpha \models h(\bar{C})$  then there is a sequence  $\bar{\alpha} = \langle \alpha_0, \dots, \alpha_m \rangle$  with  $\alpha_m = \alpha$  such that  $M, \bar{\alpha} \models \bar{C}$ .*

*Proof.* Both items are shown by a straightforward induction proof (see [26]).  $\square$

Using Def. 4 and Lem. 1, we can now show that a satisfiable pre-model remains satisfiable after removing a redundant segment.

**Lemma 2.** *Let  $\bar{\Delta} = \langle \Delta_1, \dots, \Delta_{n-1} \rangle$  be a satisfiable pre-model for  $\psi$  with redundant segment  $\bar{\Delta}_{[j+1, k]}$ . Then  $\bar{\Delta}' = \bar{\Delta}_{\leq j} \bar{\Delta}_{> k}$  is a satisfiable pre-model as well.*

*Proof.* See [26].  $\square$

It is finally possible to prove the main completeness result.

**Theorem 1** (Soundness and completeness). *Given a  $\text{LTL}_f^{\text{MT}}$  formula  $\psi$ , the tableau for  $\psi$  augmented with the PRUNE rule has an accepted branch if and only if  $\psi$  is satisfiable.*

*Proof.* As soundness is not affected by the PRUNE rule, we are only concerned with completeness. Hence, suppose  $\phi$  is satisfiable. By Prop. 3 there is a satisfiable pre-model  $\bar{\Delta} = \langle \Delta_0, \dots, \Delta_{n-1} \rangle$  for

$\phi$ . Without loss of generality, we can assume that  $\bar{\Delta}$  is of minimal length. By Prop. 4, the tableau for  $\phi$  has a corresponding branch  $\bar{u}$  with poised nodes  $\bar{\pi} = \langle \pi_0, \dots, \pi_{n-1} \rangle$  such that  $\Delta(\pi_i) = \Delta_i$  for all  $0 \leq i < n$ . As we mentioned,  $\bar{u}$  cannot have been rejected by the CONTRADICTION rule. Now, suppose by contradiction that  $\bar{u}$  has been rejected by the PRUNE rule. Then, there is a node  $\pi_i$  with  $\Gamma(\pi_i) = \Gamma(\pi_n)$  and  $h(\bar{\pi}) \models_{\mathcal{T}} h(\bar{\pi}_{\leq i})$ . But then, we have that  $\Delta_i = \Delta_n$  and  $h(\bar{\Delta}) = h(\bar{\Delta}_{\leq i})$ . That is,  $\bar{\Delta}_{[i, n]}$  is a redundant segment. By Lem. 2, we can remove it, obtaining a *shorter* satisfiable pre-model  $\bar{\Delta}_{< i}$ . But this contradicts the assumption that  $\bar{\Delta}$  was of minimal length. Hence,  $\bar{u}$  cannot have been rejected by PRUNE, and is thus an accepted branch.  $\square$

## 4 Decidable fragments

The new PRUNE rule is not capable of pruning *all* potentially infinite branches in all possible case, since  $\text{LTL}_f^{\text{MT}}$  is undecidable. However, we can identify a general sufficient condition for this to happen, given that the underlying theory  $\mathcal{T}$  is decidable (which we assume throughout this section).

**Definition 5** (Finite memory). *Given an  $\text{LTL}_f^{\text{MT}}$  formula  $\phi$ , the history set of  $\phi$  is the set of all the formulas  $h(\bar{\Delta}_{\leq i})$  for any pre-model  $\bar{\Delta}$  of  $\phi$  and any  $0 \leq i < |\bar{\Delta}|$ . A formula  $\phi$  has finite memory if its history set is finite up to  $\mathcal{T}$ -equivalence.*

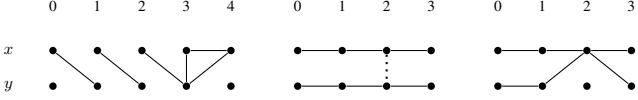
**Theorem 2** (Termination). *The tableau for an  $\text{LTL}_f^{\text{MT}}$  formula with finite memory is finite.*

*Proof.* As accepted or rejected branches are finite by definition, we are only concerned with branches that continue to expand forever without triggering any termination rule. Suppose  $\phi$  has finite memory but the tableau is infinite. Then there is at least one infinite branch since the branching degree is finite; let  $\bar{\pi} = \langle \pi_0, \pi_1, \dots \rangle$  be the poised nodes of this branch. For each prefix  $\bar{\pi}_{\leq i}$  for  $i \geq 0$ , one can check that the sequence  $\bar{\Delta} = \langle \Delta(\pi_0), \dots, \Delta(\pi_i) \rangle$  is a pre-model for  $\phi$ . Since  $\phi$  has finite memory, its history set is finite up to  $\mathcal{T}$ -equivalence. As the possible labels of tableau nodes are also finite, for some  $i$  large enough there exists a  $j < i$  such that  $\Gamma(\pi_j) = \Gamma(\pi_i)$  and  $h(\bar{\Delta}_{\leq j}) \equiv_{\mathcal{T}} h(\bar{\Delta}_{\leq i})$ , which means that  $h(\bar{\pi}_{\leq j}) \models_{\mathcal{T}} h(\bar{\pi}_{\leq i})$ . Hence the PRUNE rule would apply to  $\bar{\pi}_{\leq i}$ , contradicting the hypothesis that no termination rule is triggering along  $\bar{\pi}$ .  $\square$

While Thm. 2 gives only a semantic and, in general, undecidable condition for termination, we now show several concrete, effectively identifiable classes of  $\text{LTL}_f^{\text{MT}}$  formulas having finite memory. Indeed, we use this approach to both re-prove and extend decidability conditions previously obtained by ad-hoc methods in the literature, and to show novel results for other relevant classes of formulas.

Before giving details, we summarise our decidability results. To this end, let the set of *iteration conditions* of an  $\text{LTL}_f^{\text{MT}}$  formula  $\phi$  consist of all literals that occur in  $\phi_1$  for any subformula  $\phi_1 \cup \phi_2$  of  $\phi$ , or in  $\psi_2$  for any subformula  $\psi_1 \text{ R } \psi_2$  of  $\phi$ . We show decidability for the following classes of  $\text{LTL}_f^{\text{MT}}$  formulas:

- (NCS) Formulae without *cross-state comparisons*, i.e., that have no occurrences of  $\mathcal{V}^{\circ} \cup \mathcal{V}^{\ominus}$ , e.g.,  $(x > y \cup x + y = 2z) \wedge \text{G}(x + y > 0)$ ;
- (FX) Formulae where the only temporal operators are F, X, and  $\bar{X}$ , e.g.,  $\text{F}(p(\circ x) \wedge \bar{X}(\neg p(x))) \wedge \text{XF}(r(x, y) \vee r(\circ x, y))$ ;
- (BL) *Bounded lookback* formulae, that generalize the above two by requiring that constraint interaction via  $\mathcal{V}^{\circ}$  and  $\mathcal{V}^{\ominus}$  is restricted to finitely many configurations, e.g.,  $p(x, \circ y) \cup (\circ x = x + y)$ .



**Figure 3.** Dependency graphs for the formulas in Ex. 5 (left) and Ex. 6 (center and right). Equality edges are drawn dotted and other edges solid.

(MC) Formulas over LRA where all iteration conditions are *monotonicity constraints*, i.e., variable-to-variable or variable-to-constant comparisons. An example is the formula in Ex. 2.

(IPC) Formulas over LIA where all iteration conditions are *integer periodicity constraints*, e.g.,  $(y \equiv_3 x) \cup (x > 42) \wedge F(x+y = z)$ .

Demri and d’Souza [17, 16] showed that satisfiability is decidable for  $\text{LTL}_f^{\text{MT}}$  over arithmetics where *all* literals are monotonicity or integer periodicity constraints, but our results (MC) and (IPC) show that it suffices to restrict the shape of iteration conditions respectively. To the best of our knowledge, the result (FX) is novel; and (BL) is novel as a decidability result for satisfiability, though a similar result is known for model checking over  $\text{LTL}_f$  with arithmetic [22], and for the more restrictive condition of *feedback freedom* also supporting the theory EUF [13]. In the remainder of this section, we formally prove decidability for the five classes above.

We start with *bounded lookback* formulas. To formally define this class of formulas, we use the structure of a *dependency graph* to capture the dependencies between variables induced by a pre-model.

**Definition 6** (Dependency graph). Let  $\bar{\Delta} = \langle \Delta_0, \dots, \Delta_{n-1} \rangle$  be a pre-model. Its dependency graph is  $\text{DG}(\bar{\Delta}) = (\mathcal{V}^{\leq n}, E^=, E^{\neq})$  where  $\mathcal{V}^{\leq n} = V^0 \cup \dots \cup V^n$  is the set of nodes, and  $E^=$  and  $E^{\neq}$  are sets of two kinds of edges defined as follows.

Two variables  $x, y \in \mathcal{V}^{\leq n}$  are dependent if there is a sequence of variables  $z_0, z_1, \dots, z_m \in \mathcal{W}$  such that  $\Omega(\bar{\Delta})$  contains a literal  $\ell_0$  mentioning  $x$  and  $z_0$ , a literal  $\ell_m$  mentioning  $z_m$  and  $y$ , and, a literal  $\ell_i$  that mentions both  $z_i, z_{i+1}$  for all  $1 \leq i < m$ . In this case:

- $(x, y) \in E^=$  if all the literals  $\ell_i$  are equalities;
- $(x, y) \in E^{\neq}$  if at least one  $\ell_i$  is not an equality.

In other words,  $E^=$  is the smallest equivalence relation on  $\mathcal{V}^{\leq n}$  that contains the transitive closure of all equality literals in  $\Omega(\bar{\Delta})$ , while  $E^{\neq}$  captures connections by arbitrary other kinds of literals. Moreover, let  $\text{DG}_=(\bar{\Delta}) = (\mathcal{V}^{\leq n}, E^{\neq})$  be the graph obtained from  $\text{DG}(\bar{\Delta})$  by collapsing all equality edges to an arbitrary element in the equivalence relation induced by  $E^=$ .

**Definition 7.** For  $k \geq 0$ , an  $\text{LTL}_f^{\text{MT}}$  formula  $\psi$  has *k-bounded lookback* if for all pre-models  $\bar{\Delta}$  of  $\psi$ , it holds that all acyclic paths in  $\text{DG}_=(\bar{\Delta})$  have length at most  $k$ .

A formula has *bounded lookback* (BL) if it has  $k$ -bounded lookback for some  $k$ . The notion is an adaptation of a similar property used in model checking [22]; and as shown there, it generalizes the notion of *feedback freedom* [13] developed to verify database systems. Intuitively, bounded lookback expresses that in order to check whether a run satisfies  $\phi$ , it suffices to remember a bounded amount of information from past states. The next examples illustrate the idea.

**Example 5.** For  $\phi = p(x, \bigcirc y) \cup (\bigcirc x = x + y)$  consider the pre-model  $\bar{\Delta} = \langle \Delta_0, \Delta_0, \Delta_0, \Delta_1 \rangle$  where  $\Delta_0 = \{p(x, \bigcirc y), X\phi\}$  and  $\Delta_1 = \{\bigcirc x = x + y\}$ . We have:

$$\Omega(\bar{\Delta}) = p(x_0, y_1) \wedge p(x_1, y_2) \wedge p(x_2, y_3) \wedge x_4 = x_3 + y_3$$

Then,  $\text{DG}(\bar{\Delta})$  is pictured in Fig. 3 (left), representing all the connections between the variables  $x_0, y_0, \dots, x_4, y_4$  implied by  $\Omega(\bar{\Delta})$ .

Since there are no equality literals,  $\text{DG}_=(\bar{\Delta})$  coincides with  $\text{DG}(\bar{\Delta})$ . The longest acyclic path in  $\text{DG}_=(\bar{\Delta})$  has length 3. Though  $\phi$  has infinitely many pre-models, it can be seen that in all their DGs, acyclic paths have length  $\leq 3$ , so  $\phi$  has 3-bounded lookback.

**Example 6.** For the pre-model  $\bar{\Delta} = \langle \Delta_0, \Delta_1, \Delta_2 \rangle$  for  $\psi$  from Ex. 2, where  $\Delta_0 = \{\psi, \psi', x < 0, y = 1, \bigcirc y > y, \bigcirc x \leq x, X\psi'\}$ ,  $\Delta_1 = \{\bigcirc y > y, \bigcirc x \leq x, \psi', X\psi'\}$ , and  $\Delta_2 = \Delta_1 \cup \{x = y\}$ , we have

$$\begin{aligned} \Omega(\bar{\Delta}) = & x_0 < 0 \wedge y_0 = 1 \wedge y_1 > y_0 \wedge x_1 \leq x_0 \wedge y_2 > y_1 \\ & \wedge x_2 \leq x_1 \wedge y_3 > y_2 \wedge x_3 \leq x_2 \wedge x_2 = y_2 \end{aligned}$$

Fig. 3 shows  $\text{DG}(\bar{\Delta})$  (center) and  $\text{DG}_=(\bar{\Delta})$  (right). The longest path in  $\text{DG}_=(\bar{\Delta})$  has length 4. However,  $\psi$  has infinitely many pre-models  $\bar{\Delta}_m = \langle \Delta_0, \Delta_1, \dots, \Delta_1, \Delta_2 \rangle$  with  $m$  repetitions of  $\Delta_1$ , for any  $m \geq 0$ , which have similar  $\text{DG}_=$ ’s with paths of length  $2(m+1)$ . So  $\psi$  does not have  $k$ -bounded lookback, for any  $k$ .

The proof of the following result recasts the approach from [22, Thm. 5.10] for pre-models and satisfiability.

**Theorem 3.** Satisfiability of BL formulas is decidable.

*Proof.* Let  $\phi$  have  $k$ -bounded lookback, and  $\bar{\Delta}$  a pre-model of length  $n$  for  $\phi$ . The history constraint  $h(\bar{\Delta})$  encodes  $\text{DG}(\bar{\Delta})$ . Let  $\chi$  be the formula obtained from  $h(\bar{\Delta})$  by removing all equalities between variables and replacing each variable in  $\mathcal{V}^{\leq n}$  by a representative from its  $E^=$ -equivalence class. Then  $\chi \equiv_{\mathcal{T}} h(\bar{\Delta})$  and  $\chi$  encodes  $\text{DG}_=(\bar{\Delta})$ . Since all acyclic paths in  $\text{DG}_=(\bar{\Delta})$  have length at most  $k$ , each variable in  $V^n$  is connected in  $\text{DG}_=(\bar{\Delta})$  to at most  $k$  variables in  $\mathcal{V}^{\leq n}$ . As  $\chi$  encodes  $\text{DG}_=(\bar{\Delta})$ ,  $\chi$  is equivalent to a formula with at most  $k \cdot |\mathcal{V}|$  quantified variables. All literals in  $\chi$  are (renamed) first-order formulas in  $\phi$ . The number of formulas with a bounded number of quantifiers and finite vocabulary is finite up to equivalence, so  $\psi$  has finite memory, and by Thm. 2, the tableau is finite.  $\square$

Note that for a given  $k$  and  $\text{LTL}_f^{\text{MT}}$  formula  $\psi$ , it is decidable whether  $\psi$  has  $k$ -bounded lookback, by checking whether none of the finitely many (prefixes of) pre-models of length  $k+1$  has a path in  $\text{DG}_=$  of length more than  $k$  (cf., [22]). However, it is undecidable whether there is *some*  $k$  such that  $\psi$  has  $k$ -bounded lookback.

Let a formula have *cross-state comparisons* if it contains variables in  $\mathcal{V}^{\bigcirc}$  or  $\mathcal{V}^{\bigcirc\bigcirc}$ . Note that for formulas without cross-state comparisons, dependency graphs have only edges from some  $x_i$  to some  $y_i$  for the same  $i$  (i.e., vertical edges if pictured as in Fig. 3), so all acyclic paths have length at most  $|\mathcal{V}|$ . We hence obtain the following:

**Corollary 1.** Satisfiability of formulas without cross-state comparisons is decidable.

Now, let an  $\text{LTL}_f^{\text{MT}}$  formula be an FX formula if its only temporal operators are F, X, and  $\tilde{X}$ .

**Theorem 4.** Satisfiability of FX formulas is decidable.

*Proof.* Suppose an FX formula  $\phi$  contains  $m$  literals, and let  $\bar{\Delta} = \langle \Delta_0, \dots, \Delta_{n-1} \rangle$  be a pre-model for it. By the expansion rules of the F, X, and  $\tilde{X}$  operators, and the minimality of atoms, every literal occurrence in  $\phi$  corresponds to at most one occurrence in the pre-model. Thus,  $\bar{\Delta}$  contains at most  $m$  literals overall, and each path in its dependency graph is upper-bounded by  $m \cdot |\mathcal{V}|$ , hence  $\phi$  has bounded lookback. The claim then follows from Thm. 3.  $\square$

We next consider fragments of  $\text{LTL}_f^{\text{MT}}$  over arithmetic theories. *Monotonicity constraints* (MC) restrict linear arithmetics over the rationals, demanding all constraints to be of the form  $p \odot q$  where  $p, q \in \mathbb{Q} \cup \mathbb{V} \cup \mathbb{V}^\circ \cup \mathbb{V}^\ominus$  and  $\odot \in \{=, \neq, \leq, <\}$ . An  $\text{LTL}_f^{\text{MT}}$  formula  $\phi$  is an *MC formula* if all literals in  $\phi$  are MCs, such as in the formula from Ex. 2. Satisfiability of MC formulas is known to be decidable [17, Cor. 5.5]. Here, we prove decidability for a larger class.

**Definition 8** (Quasi-MC formulas). *An  $\text{LTL}_f^{\text{MT}}$  formula over the signature of LRA is quasi-MC if all its iteration conditions are MCs.*

E.g.,  $(\bigcirc x > x \wedge \bigcirc y > y) \cup (x+y > 10)$  is not an MC-, but a quasi-MC formula. MC formulas are important in BPM, as they can model decision tables [15]. To show decidability of quasi-MC formulas, we use the following fact about quantifier elimination [33, Sec. 5.4]: if  $\phi$  is an LRA formula where all literals are MCs over a set of constants  $\mathcal{K}$  and variables  $X \cup \{x\}$ , then one can compute a formula  $\phi' \equiv_{\text{LRA}} \exists x. \phi$  such that all literals in  $\phi'$  are MCs over constants  $\mathcal{K}$  and variables  $X$ ; e.g., using a Fourier-Motzkin procedure.

**Theorem 5.** *Satisfiability of quasi-MC formulas is decidable.*

*Proof.* Let  $\mathcal{K}$  be the set of constants,  $I$  the set of iteration conditions,  $A$  the set of all first-order formulas in a quasi-MC formula  $\phi$ , and  $m$  the number of occurrences of formulas of  $A$  in  $\phi$ . For a pre-model  $\bar{\Delta} = \langle \Delta_0, \dots, \Delta_{n-1} \rangle$ , let  $J = \{i_1, \dots, i_k\} \subseteq \{0, \dots, n-1\}$  be all indices such that  $F(\Delta_{i_j})$  contains a formula in  $A \setminus I$ . *W.l.o.g.*, assume that  $n-1 \in J$ ; otherwise the reasoning is similar. Note that  $k \leq m$  since every occurrence of a first-order formula in  $\phi$  that is not an iteration condition can occur in at most one atom in a pre-model. Now,  $\Omega(\bar{\Delta})$  has free variables  $\mathcal{V}^{\leq n} = V^0 \cup \dots \cup V^n$ ; let  $X \subseteq \mathcal{V}^{\leq n}$  be the set of variables occurring in  $\{F(\Delta_j)^{(j)} \mid j \in J\}$ , and  $Y = \mathcal{V}^{\leq n} \setminus X$ . Then we can write  $h(\bar{\Delta})$  as

$$\left( \exists X. (\exists Y. \bigwedge_{i \in N \setminus J} C_i^{(i)} \wedge \bigwedge_{i \in J \setminus \{m-1\}} C_i^{(i)} \wedge L(C_{m-1})^{(m-1)}) \right) [\bar{V}]$$

where  $C_i = F(\Delta_i)$ . By the QE property of MCs, the subformula  $\exists Y. \bigwedge_{i \in N \setminus J} C_i^{(i)}$  is LRA-equivalent to a first-order formula  $\chi$  where all literals are MCs over constants  $\mathcal{K}$  and variables  $\mathbb{V} \cup X$ . There are only finitely many such  $\chi$  up to equivalence, as there are only finitely many MCs over a finite set of variables and constants. Moreover, the number of possibilities for the sequence  $C_{i_1}, \dots, C_{i_k}$  is bounded by  $2^{2^m}$  since all these  $C_{i_j}$  must be conjunctions of subsets of  $A \setminus I$ , and  $k \leq m$ . Thus, up to equivalence, there are finitely many possibilities for  $h(\bar{\Delta})$ , so the history set is finite.  $\square$

*Integer periodicity constraints* (IPCs) restrict linear *integer arithmetic* (LIA) and are, e.g., used in calendar formalisms [16]. Precisely, IPC atoms have the form  $x = y$  or  $x \odot d$  for  $\odot \in \{=, \neq, <, >, \equiv_k\}$ , or  $x \equiv_k y + d$ , for variables  $x, y$  with domain  $\mathbb{Z}$  and  $k, d \in \mathbb{N}$ . An  $\text{LTL}_f^{\text{MT}}$  formula  $\phi$  over LIA is an *IPC formula* if all first-order formulas in  $\phi$  are IPCs, and a *quasi-IPC formula* if all iteration conditions are IPCs. IPC formulas are known to be decidable [16, Thm. 3].

We extend this result to quasi-IPC formulas by using a quantifier elimination property as for MCs: if  $\phi$  is a first-order formula where all literals are IPCs over a set of constants  $\mathcal{K}$  and variables  $X \cup \{x\}$ , then one can compute a formula  $\phi' \equiv_{\text{LIA}} \exists x. \phi$  such that  $\phi'$  is a first-order formula where all literals are IPCs over constants  $\mathcal{K}$  and variables  $X$  [16, Thm. 2]. Then, the following can be proven exactly like Thm. 5, using the fact that there are only finitely many LIA formulas where all literals are IPCs over finite sets of variables and constants:

**Theorem 6.** *Satisfiability of quasi-IPC formulas is decidable.*

## 5 Related work and conclusions

In this paper we considered the satisfiability problem for  $\text{LTL}_f^{\text{MT}}$ , a highly expressive extension of  $\text{LTL}_f$ . In earlier work, a tableau system for  $\text{LTL}_f^{\text{MT}}$  was proposed that is, however, incomplete to show unsatisfiability. In this paper, we proposed a pruning rule for this tableau that we proved sound and complete. We show that the tableau construction terminates whenever the  $\text{LTL}_f^{\text{MT}}$  formula satisfies the semantic property of finite memory, and use this abstract termination condition to prove decidability for several concrete, checkable, and relevant classes of formulas, extending results from the literature.

Given the limited expressivity of propositional LTL, several extensions with richer background theories have been considered, in particular (fragments of) arithmetic theories [17, 16, 13, 19]. The extension of LTL with first-order theories is highly challenging, as even the most basic verification tasks become undecidable [4]. A starting point for this work is the  $\text{LTL}_f^{\text{MT}}$  tableau by Geatti *et al.* [24], which provides a semi-decision procedure; but, lacking a pruning rule, is rarely able to show unsatisfiability, and no decidability results for fragments of  $\text{LTL}_f^{\text{MT}}$  are given. However, some decidability results for model checking and satisfiability (which are equivalent in linear-time temporal logics) for LTL with more specific theories are known. Demri and D'Souza [17] showed that satisfiability of LTL with monotonicity constraints (MCs), over both integers and rationals, is decidable in PSPACE, and the same holds for LTL over integer periodicity constraints [16]. Our results for the (MC) and (IPC) fragments strictly extend these decidability results, since we only restrict iteration conditions of formulas. The picture gets more diverse for branching-time temporal logics equipped with similar arithmetic theories; in this case, satisfiability and model checking do no longer coincide [11, 10, 23, 21]. Damaggio *et al.* [13] considered LTL model checking for transition systems that operate over databases and include arithmetic conditions, and proved decidability if the system together with the LTL formula satisfies the property of *feedback freedom*. For purely arithmetic transition systems, feedback freedom was extended by Felli *et al.* to that of *bounded lookback* [22]. Our decidability result for (BL) takes this idea to arbitrary theories, and recasts it for the satisfiability problem, thus strictly extending [13, 22]. We showed that in the context of satisfiability, (BL) implies decidability of the (FX) fragment, which has no counterpart in model checking. Deutsch *et al.* [19] proved decidability of model checking for hierarchic transition systems and a variant of first-order LTL (HLTL-FO), but this logic is in general incomparable to  $\text{LTL}_f^{\text{MT}}$ . Our notion of *history constraints* is inspired by the respective notions from [22, 13, 7], though we recast it here for satisfiability and in the setting of a tableau system.

Tableau systems for LTL and extensions thereof have been extensively considered [35, 39, 37, 29]. The tableau for  $\text{LTL}_f^{\text{MT}}$  provided in [24] is based on Reynolds' one-pass and tree-shaped tableau for LTL [37], whose PRUNE rule does not transfer directly to the first-order case. Tableau calculi for first-order extensions of LTL have also been proposed [32], but they are not parameterised over the underlying theory, and the considered logic do not support  $\bigcirc$  and  $\ominus$  terms.

Several directions for future work can be considered. Following the path taken by [24], an SMT encoding of our PRUNE rule would allow for its implementation in the BLACK temporal reasoning framework [27]. Moreover, whether these results can be extended to a version of  $\text{LTL}_f^{\text{MT}}$  supporting time-varying relations is still open. Finally, we want to study also other, related tasks such as branching-time logics modulo theories, and  $\text{LTL}_f^{\text{MT}}$  monitoring [20].

## Acknowledgements

This work was partially funded by the UNIBZ project ADAPTERS, and the PRIN MIUR project PINPOINT Prot. 2020FNEB27. Nicola Gigante acknowledges the support of the PURPLE project, 1st Open Call for Innovators of the AIPlan4EU H2020 project, a project funded by EU Horizon 2020 research and innovation programme under GA n. 101016442 (since 2021). Luca Geatti acknowledges the support from the 2022 Italian INdAM-GNCS project “*Elaborazione del Linguaggio Naturale e Logica Temporale per la Formalizzazione di Testi*”, ref. no. CUP\_E55F22000270001.

## References

- [1] Fahiem Bacchus and Froduald Kabanza, ‘Using temporal logics to express search control knowledge for planning’, *Artif. Intell.*, **116**(1-2), 123–191, (2000).
- [2] Clark W. Barrett, Roberto Sebastiani, Sanjit A. Seshia, and Cesare Tinelli, ‘Satisfiability modulo theories’, in *Handbook of Satisfiability - Second Edition*, eds., Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, volume 336 of *Frontiers in Artificial Intelligence and Applications*, 1267–1329, IOS Press, (2021).
- [3] Diego Calvanese, Giuseppe de Giacomo, Marco Montali, and Fabio Patrizi, ‘First-order  $\mu$ -calculus over generic transition systems and applications to the situation calculus’, *Inform. Comput.*, **259**(3), 328–347, (2018).
- [4] Diego Calvanese, Giuseppe De Giacomo, Marco Montali, and Fabio Patrizi, ‘Verification and monitoring for first-order LTL with persistence-preserving quantification over finite and infinite traces’, in *Proc. 31st IJCAI*, pp. 2553–2560, (2022).
- [5] Diego Calvanese, Silvio Ghilardi, Alessandro Gianola, Marco Montali, and Andrey Rivkin, ‘Formal modeling and SMT-based parameterized verification of data-aware BPMN’, in *Proc. of BPM 2019*, volume 11675 of *LNCS*, pp. 157–175, (2019).
- [6] Diego Calvanese, Silvio Ghilardi, Alessandro Gianola, Marco Montali, and Andrey Rivkin, ‘SMT-based verification of data-aware processes: a model-theoretic approach’, *Math. Struct. Comput. Sci.*, **30**(3), 271–313, (2020).
- [7] Diego Calvanese, Silvio Ghilardi, Alessandro Gianola, Marco Montali, and Andrey Rivkin, ‘Model completeness, uniform interpolants and superposition calculus’, *J. Autom. Reason.*, **65**(7), 941–969, (2021).
- [8] Diego Calvanese, Giuseppe De Giacomo, and Marco Montali, ‘Foundations of data-aware process analysis: a database theory perspective’, in *Proc. of PODS 2013*, pp. 1–12, ACM, (2013).
- [9] Alberto Camacho, Jorge A. Baier, Christian J. Muise, and Sheila A. McIlraith, ‘Finite LTL synthesis as planning’, in *Proc. 28th ICAPS*, pp. 29–38, (2018).
- [10] C. Carapelle, A. Kartzow, and M. Lohrey, ‘Satisfiability of ECTL\* with constraints’, *Journal of Computer and System Sciences*, **82**(5), 826–855, (2016).
- [11] Karlis Cerans, ‘Deciding properties of integral relational automata’, in *Proc. 21st ICALP*, volume 820 of *LNCS*, pp. 35–46, (1994).
- [12] Alessandro Cimatti, Alberto Griggio, Enrico Magnago, Marco Roveri, and Stefano Tonetta, ‘SMT-based satisfiability of first-order LTL with event freezing functions and metric operators’, *Inf. Comput.*, **272**, 104502, (2020).
- [13] Elio Damaggio, Alin Deutsch, and Victor Vianu, ‘Artifact systems with data dependencies and arithmetic’, *ACM Trans. Database Syst.*, **37**(3), 22:1–22:36, (2012).
- [14] Giuseppe De Giacomo and Moshe Y. Vardi, ‘Linear temporal logic and linear dynamic logic on finite traces’, in *Proc. 23rd IJCAI*, pp. 854–860, (2013).
- [15] Massimiliano de Leoni, Paolo Felli, and Marco Montali, ‘Integrating BPMN and DMN: modeling and analysis’, *J. Data Semant.*, **10**(1), 165–188, (2021).
- [16] Stéphane Demri, ‘LTL over integer periodicity constraints’, *Theor. Comput. Sci.*, **360**(1-3), 96–123, (2006).
- [17] Stéphane Demri and Deepak D’Souza, ‘An automata-theoretic approach to constraint LTL’, *Inform. Comput.*, **205**(3), 380–415, (2007).
- [18] Alin Deutsch, Yuliang Li, and Victor Vianu, ‘Verification of hierarchical artifact systems’, in *Proc. of PODS 2016*, pp. 179–194, ACM, (2016).
- [19] Alin Deutsch, Yuliang Li, and Victor Vianu, ‘Verification of hierarchical artifact systems’, *ACM Trans. Database Syst.*, **44**(3), 12:1–12:68, (2019).
- [20] Paolo Felli, Marco Montali, Fabio Patrizi, and Sarah Winkler, ‘Monitoring arithmetic temporal properties on finite traces’, in *Proc. 35th AAAI*, pp. 6346–6354, (2023).
- [21] Paolo Felli, Marco Montali, and Sarah Winkler, ‘CTL\* model checking for data-aware dynamic systems with arithmetic’, in *Proc. 11th IJCAR*, volume 13385, pp. 36–56, (2022).
- [22] Paolo Felli, Marco Montali, and Sarah Winkler, ‘Linear-time verification of data-aware dynamic systems with arithmetic’, in *Proc. 34th AAAI*, pp. 5642–5650, (2022).
- [23] Régis Gascon, ‘An automata-based approach for CTL\* with constraints’, in *Proc. INFINITY 2006, 2007 and 2008*, volume 239, pp. 193–211, (2009).
- [24] Luca Geatti, Alessandro Gianola, and Nicola Gigante, ‘Linear temporal logic modulo theories over finite traces’, in *Proc. 31st IJCAI*, pp. 2641–2647, (2022).
- [25] Luca Geatti, Alessandro Gianola, and Nicola Gigante, ‘Linear temporal logic modulo theories over finite traces (extended version)’, *CoRR*, **abs/2204.13693**, (2022).
- [26] Luca Geatti, Alessandro Gianola, Nicola Gigante, and Sarah Winkler, ‘Decidable fragments of LTLf modulo theories (extended version)’, *CoRR*, **abs/2307.16840**, (2023).
- [27] Luca Geatti, Nicola Gigante, and Angelo Montanari, ‘A SAT-based encoding of the one-pass and tree-shaped tableau system for LTL’, in *Proc. 28th TABLEAUX*, volume 11714 of *LNCS*, pp. 3–20, (2019).
- [28] Luca Geatti, Nicola Gigante, Angelo Montanari, and Mark Reynolds, ‘One-pass and tree-shaped tableau systems for TPTL and TPTLb+Past’, *Inform. Comput.*, **278**, 104599, (2021).
- [29] Luca Geatti, Nicola Gigante, Angelo Montanari, and Mark Reynolds, ‘One-pass and tree-shaped tableau systems for TPTL and TPTLb+Past’, *Inform. Comput.*, (2021). in press.
- [30] Silvio Ghilardi, Alessandro Gianola, Marco Montali, and Andrey Rivkin, ‘Petri net-based object-centric processes with read-only data’, *Inf. Syst.*, **107**, 102011, (2022).
- [31] Giuseppe De Giacomo, Riccardo De Masellis, Marco Grasso, Fabrizio Maria Maggi, and Marco Montali, ‘Monitoring business metaconstraints based on LTL and LDL for finite traces’, in *Proc. of BPM 2014*, volume 8659 of *LNCS*, pp. 1–17, (2014).
- [32] Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyashev, ‘Temporalising tableaux’, *Stud Logica*, **76**(1), 91–134, (2004).
- [33] Daniel Kroening and Ofer Strichman, *Decision Procedures – An Algorithmic Point of View, Second Edition*, Springer, 2016.
- [34] Jianwen Li, Geguang Pu, Yueling Zhang, Moshe Y. Vardi, and Kristin Y. Rozier, ‘SAT-based explicit LTLf satisfiability checking’, *Artif. Intell.*, **289**, 103369, (2020).
- [35] Orna Lichtenstein and Amir Pnueli, ‘Propositional Temporal Logics: Decidability and Completeness’, *Logic Journal of the IGPL*, **8**(1), 55–85, (2000).
- [36] Amir Pnueli, ‘The temporal logic of programs’, in *18th Annual Symposium on Foundations of Computer Science*, pp. 46–57, IEEE Computer Society, (1977).
- [37] Mark Reynolds, ‘A New Rule for LTL Tableaux’, in *Proc. of the 7th International Symposium on Games, Automata, Logics and Formal Verification*, volume 226 of *EPTCS*, pp. 287–301, (2016).
- [38] Kristin Y. Rozier and Moshe Y. Vardi, ‘LTL satisfiability checking’, *Int. J. Softw. Tools Technol. Transf.*, **12**(2), 123–137, (2010).
- [39] S. Schwendimann, ‘A New One-Pass Tableau Calculus for PLTL’, in *Proc. 7th TABLEAUX*, volume 1397 of *LNCS*, pp. 277–292, (1998).



## A Proofs

**Lemma 1.** Let  $M$  be a  $\Sigma$ -structure and  $\bar{C} = \langle C_0, \dots, C_{m-1} \rangle$  be a sequence of first-order formulas with free variables  $\mathcal{V} \cup \mathcal{V}^\circ \cup \mathcal{V}^\otimes$ , for  $m \geq 1$ .

- (1) If  $M, \langle \alpha_0, \dots, \alpha_m \rangle \models \bar{C}$  then  $M, \alpha_m \models h(\bar{C})$ .
- (2) If  $M, \alpha \models h(\bar{C})$  then there is a sequence  $\bar{\alpha} = \langle \alpha_0, \dots, \alpha_m \rangle$  with  $\alpha_m = \alpha$  such that  $M, \bar{\alpha} \models \bar{C}$ .

*Proof.* Both items are by induction on  $m$ .

- (1) If  $m = 1$  and  $M, \langle \alpha_0, \alpha_1 \rangle \models \langle C_0 \rangle$  then  $M, \alpha_0 \otimes \alpha_1 \models L(C_0)$ , so after renaming and quantification,

$$M, \alpha_1 \models (\exists V^0. L(C_0)^{(0)})[\bar{\mathcal{V}}] = h(\langle C_0 \rangle).$$

For the induction step, suppose  $\bar{C} = \langle C_0, \dots, C_m \rangle$  and  $M, \langle \alpha_0, \dots, \alpha_{m+1} \rangle \models \bar{C}$ . Let  $M'$  be like  $M$  but such that  $M' \models \ell$ . For  $\bar{C}' = \langle C_0, \dots, C_{m-1} \rangle$ , we have  $M', \langle \alpha_0, \dots, \alpha_m \rangle \models \bar{C}'$ . By the induction hypothesis,  $M', \alpha_m \models h(\bar{C}')$ . Since  $M' \models \ell$ , it also holds that  $M', \alpha_m \models (\exists V^0 \dots V^{m-1}. \bigwedge_{i=0}^{m-1} C_i^{(i)})[\bar{\mathcal{V}}]$ , i.e.,  $M'$  and  $\alpha_m$  satisfy the formula that is like  $h(\bar{C}')$  but where  $L$  is not applied to  $C_{m-1}$ ; call this fact  $(\star)$ . Let  $\alpha'_m$  be the substitution with domain  $V^m$  such that  $\alpha'_m(v^m) = \alpha_m(v)$  and  $\alpha'_{m+1}$  have domain  $V_{m+1}$  such that  $\alpha'_{m+1}(v^{m+1}) = \alpha_{m+1}(v)$  for all  $v \in \mathcal{V}$ , so they are like  $\alpha_m$  and  $\alpha_{m+1}$ , respectively, but with domains  $V^m$  and  $V^{m+1}$ . Since  $M, \langle \alpha_0, \dots, \alpha_{m+1} \rangle \models \bar{C}$ , we have  $M, \alpha_m \otimes \alpha_{m+1} \models L(C_m)$ , so  $M, \alpha'_m \cup \alpha'_{m+1} \models L(C_m)^{(m)}$ . From  $(\star)$  we have  $M, \alpha'_m \models \exists V^0 \dots V^{m-1}. \bigwedge_{i=0}^{m-1} C_i^{(i)}$  (using  $M$  instead of  $M'$ , as  $\ell$  is not involved). By combining this with the above, we have  $M, \alpha'_m \cup \alpha'_{m+1} \models \exists V^0 \dots V^{m-1}. \bigwedge_{i=0}^{m-1} C_i^{(i)} \wedge L(C_m)^m$ , so  $M, \alpha'_{m+1} \models \exists V^0 \dots V^m. \bigwedge_{i=0}^{m-1} C_i^{(i)} \wedge L(C_m)^m$ , hence by renaming variables,  $M, \alpha_m \models h(\bar{C})$ .

- (2) Let  $m = 1$  and  $M, \alpha \models h(\langle C_0 \rangle)$ , which means  $M, \alpha \models (\exists V^0. L(C_0)^{(0)})[\bar{\mathcal{V}}]$ . Let  $\alpha'_1$  have domain  $V^1$  such that  $\alpha'_1(v^1) = \alpha(v)$  for all  $v \in \mathcal{V}$ . There must be an assignment  $\alpha'_0$  with domain  $V^0$  such that  $M, \alpha'_0 \cup \alpha'_1 \models L(C_0)^{(0)}$ , so for  $\alpha_0$  with domain  $V$  such that  $\alpha'_0(v^0) = \alpha_0(v)$  for all  $v \in V$ , it holds that  $M, \alpha_0 \otimes \alpha \models L(C_0)$ , so  $M, \langle \alpha_0, \alpha \rangle \models \langle C_0 \rangle$ .

For the induction step, let  $\bar{C} = \langle C_0, \dots, C_m \rangle$ ,  $\bar{C}' = \langle C_0, \dots, C_{m-1} \rangle$ , and suppose  $M, \alpha \models h(\bar{C})$ , so

$$M, \alpha \models (\exists V^0 \dots V^m. \bigwedge_{i=0}^{m-1} C_i^{(i)} \wedge L(C_m)^m)[\bar{\mathcal{V}}]$$

Let  $\hat{\alpha}$  have domain  $V^{m+1}$  such that  $\hat{\alpha}(v^{m+1}) = \alpha(v)$  for all  $v \in \mathcal{V}$ , so  $M, \hat{\alpha} \models \exists V^0 \dots V^m. \bigwedge_{i=0}^{m-1} C_i^{(i)} \wedge L(C_m)^m$ . Thus there is an assignment  $\hat{\alpha}'$  with domain  $V^m$  such that  $M, \hat{\alpha} \cup \hat{\alpha}' \models \exists V^0 \dots V^{m-1}. \bigwedge_{i=0}^{m-1} C_i^{(i)} \wedge L(C_m)^m$   $(\star)$ . For  $\alpha'$  with domain  $\mathcal{V}$  such that  $\hat{\alpha}'(v^m) = \alpha'(v)$  for all  $v \in \mathcal{V}$ , it thus holds that  $M, \alpha' \models \exists V^0 \dots V^{m-1}. \bigwedge_{i=0}^{m-1} C_i^{(i)}$ . Let  $M'$  be like  $M$  but such that  $M' \models \ell$ . We have  $M', \alpha' \models (\exists V^0 \dots V^{m-1}. \bigwedge_{i=0}^{m-1} C_i^{(i)} \wedge L(C_{m-1})^{m-1})[\bar{\mathcal{V}}] = h(\bar{C}')$ . By the induction hypothesis, there is a sequence  $\langle \alpha_0, \dots, \alpha_m \rangle$  such that  $M', \langle \alpha_0, \dots, \alpha_m \rangle \models \bar{C}'$  and  $\alpha_m = \alpha'$ . Since  $M' \models \ell$ , by definition of  $L$ , it holds that  $M, \alpha_i \otimes \alpha_{i+1} \models C_i$  for all  $0 \leq i < m$  (where  $C_{m-1}$  is not modified by  $L$ ). From  $(\star)$ , we also have  $M, \alpha' \otimes \alpha \models L(C_m)$ , so for  $\bar{\alpha} = \langle \alpha_0, \dots, \alpha_m, \alpha \rangle$  we have  $M, \bar{\alpha} \models \bar{C}$ .  $\square$

**Lemma 2.** Let  $\bar{\Delta} = \langle \Delta_1, \dots, \Delta_{n-1} \rangle$  be a satisfiable pre-model for  $\psi$  with redundant segment  $\bar{\Delta}_{[j+1, k]}$ . Then  $\bar{\Delta}' = \bar{\Delta}_{\leq j} \bar{\Delta}_{> k}$  is a satisfiable pre-model as well.

*Proof.* First, we show that,  $\bar{\Delta}'$  is still a pre-model for  $\phi$ : Since  $\Delta_j = \bar{\Delta}_k$ , for every  $X\psi \in \Delta_j$  it must hold that  $\psi \in \Delta_{k+1}$ ; and for every  $X\psi \in \Delta_j$ , there is nothing to show if  $k = n$ , or otherwise  $\psi \in \Delta_{k+1}$  must hold as well. If  $\psi_1 \cup \psi_2 \in \Delta_j$  then  $\psi_1 \cup \psi_2 \in \Delta_k$ , so the eventuality must be fulfilled at a later point, and similarly for  $R$ . Minimality with respect to set inclusion is clear.

It remains to show that  $\bar{\Delta}'$  is satisfiable. We abbreviate the first-order formulas in  $\Delta_i$  by  $\phi_i := \bigwedge F(\Delta_i)$  for all  $0 \leq i < n$ . By assumption,  $\bar{\Delta}$  is satisfiable, so  $\Omega(\langle \phi_1, \dots, \phi_{n-1} \rangle) \wedge \neg \ell$  is  $\mathcal{T}$ -satisfiable. Thus also  $h(\bar{\Delta}) \wedge \neg \ell$  is  $\mathcal{T}$ -satisfiable, so there are a  $\Sigma$ -structure  $M$  and a state variable assignment  $\alpha$  such that  $M, \alpha \models h(\bar{\Delta}) \wedge \neg \ell$   $(\star)$ . By Lem. 1 there is a sequence  $\bar{\alpha} = \langle \alpha_0, \dots, \alpha_n \rangle$  such that  $\alpha_n = \alpha$  and  $M, \bar{\alpha} \models \langle \phi_1, \dots, \phi_n \rangle$ . Let  $M'$  be like  $M$  except that  $M' \models \ell$ . Then  $M', \langle \alpha_0, \dots, \alpha_k \rangle \models \langle \phi_1, \dots, \phi_k \rangle \wedge \ell$ . By Lem. 1 it thus holds that  $M', \alpha_k \models h(\bar{\Delta}_{\leq k})$ . Since  $h(\bar{\Delta}_{\leq k}) \models \neg \tau$   $h(\bar{\Delta}_{\leq j})$ , it holds that  $M', \alpha_k \models h(\bar{\Delta}_{\leq j})$ .

Again by Lem. 1 there is a sequence  $\bar{\alpha}' = \langle \alpha'_0, \dots, \alpha'_j \rangle$  such that  $\alpha'_j = \alpha_k$  and  $M', \bar{\alpha}' \models \langle \phi_0, \dots, \phi_j \rangle$ . Since  $M' \models \ell$ , we have  $M', \alpha_i \otimes \alpha_{i+1} \models \phi_i$  for all  $0 \leq i < j$ . With  $\alpha'_j = \alpha_k$ , it follows that the combined sequence  $\bar{\alpha}'' = \langle \alpha'_0, \dots, \alpha'_{j-1}, \alpha_k, \dots, \alpha_n \rangle$  satisfies  $M', \bar{\alpha}'' \models \langle \phi_1, \dots, \phi_{j-1}, \phi_k, \dots, \phi_n \rangle$ . Again by Lem. 1,  $h(\bar{\Delta}')$  is  $\mathcal{T}$ -satisfiable. Finally,  $M, \alpha_n \models h(\bar{\Delta}') \wedge \neg \ell$  must hold because  $M, \alpha_{n-1} \otimes \alpha_n \models \phi_n \wedge \neg \ell$  follows from  $(\star)$ .  $\square$