Automata-based Computation of Temporal Equilibrium Models*

Pedro Cabalar[1] and Stéphane Demri[2]

[1] Department of Computer Science,
University of Corunna, Spain.
cabalar@udc.es
[2] LSV, ENS de Cachan, CNRS, INRIA, France.
demri@lsv.ens-cachan.fr

Abstract. Temporal Equilibrium Logic (TEL) is a formalism for temporal logic programming that generalizes the paradigm of Answer Set Programming (ASP) introducing modal temporal operators from standard Linear-time Temporal Logic (LTL). In this paper we solve some problems that remained open for TEL like decidability, bounds for computational complexity as well as computation of temporal equilibrium models for arbitrary theories. We propose a method for the latter that consists in building a Büchi automaton that accepts exactly the temporal equilibrium models of a given theory, providing an automata-based decision procedure and illustrating the ω-regularity of such sets. We show that TEL satisfiability can be solved in exponential space and it is hard for polynomial space. Finally, given two theories, we provide a decision procedure to check if they have the same temporal equilibrium models.

1 Introduction

Stable models. Stable models have their roots in Logic Programming and in the search for a semantical interpretation of default negation [9] (or *answer set* semantics). They have given rise to a successful declarative paradigm, known as *Answer Set Programming* (ASP) [18,15], for practical knowledge representation. ASP has been applied to a wide spectrum of domains for solving several types of reasoning tasks: making diagnosis for the Space Shuttle [19], information integration of different data sources [13], distributing seaport employees in work teams [10] or automated music composition [3] to cite some examples. Some of these application scenarios frequently involve representing transition-based systems under linear time, so that discrete instants are identified with natural numbers. ASP offers interesting features for a formal treatment of temporal scenarios. For instance, it provides a high degree of *elaboration tolerance* [16], allowing a simple and natural solution to typical representational issues such as the *frame* problem and the *ramification* problem, see respectively [17] and [12].

* This research was partially supported by Spanish MEC project TIN2009-14562-C05-04 and Xunta de Galicia project INCITE08-PXIB105159PR.

Another interesting feature is that it allows a uniform treatment of different kinds of reasoning problems such as prediction, postdiction, planning, diagnosis or verification. However, since ASP is not a temporal formalism, it also involves some difficulties for dealing with temporal problems. In particular, since most ASP tools must deal with finite domains, this additionally requires fixing a finite path length with an obvious impossibility for solving problems such as proving the non-existence of a plan for a given planning scenario, or checking whether two temporal representations are *strongly equivalent* (i.e., they are interchangeable inside any context and for any path length).

Temporal Equilibrium Logic. To overcome these difficulties, in [1] a temporal extension of ASP, called *Temporal Equilibrium Logic* (TEL), was considered. This extension is an orthogonal combination of *linear-time temporal logic* (LTL) (see e.g. [22]) with the nonmonotonic formalism of *Equilibrium Logic* [20], probably the most general and best studied logical characterisation of ASP. TEL extends the stable model semantics to arbitrary LTL theories, that is, sets of formulae that combine atoms, the standard Boolean connectives, and the temporal operators X (read "next"), G (read "always"), F (read "eventually"), U (read "until") and R (read "release").

Towards arbitrary TEL theories. The definition of TEL has allowed studying problems like the aforementioned strong equivalence [1] of two temporal theories, but it had mostly remained as a theoretical tool, since there was no method for computing the temporal stable models of a temporal theory, at least until quite recently. In a first step in this direction, the paper [2] started from the normal form for TEL called *temporal logic programs* (TLPs) from [5] and showed that, when a syntactic subclass is considered (the so-called *splitable* TLPs), its temporal stable models can be obtained by a translation into LTL. This method has been implemented in a tool called `STeLP` [6] that uses an LTL model checker as a backend and provides the temporal stable models of a splitable TLP in terms of a Büchi automaton.

Although the splitable TLPs are expressive enough to capture most temporal scenarios treated in the ASP literature, a general method to compute the temporal equilibrium models for *arbitrary* TEL theories was not available until now. The interest for obtaining such a method is not only to cover the full expressiveness of this logic, but also to show its decidability and assess the complexity associated to its main reasoning tasks. In this sense, it is not convenient to use TLPs as a starting point since, despite of being a normal form for TEL, they are obtained by introducing new auxiliary atoms not present in the original propositional signature.

Our contributions. In this paper we cover this gap and introduce an automata-based method to compute the temporal equilibrium models of an arbitrary temporal theory. We will pay a special attention to recall standard relationships between LTL and Büchi automata in order to facilitate the connection between ASP concepts and those from model-checking with temporal logics. More precisely, we propose automata-based decision procedures as follows:

1. We show that the satisfiability problem for the monotonic basis of TEL – the so-called logic of *Temporal Here-and-There* (THT) – can be solved in PSPACE by translation into the satisfiability problem for LTL. Whence, any decision procedure for LTL (automata-based, tableaux-based, resolution-based, etc.) can be used for THT. We are also able to demonstrate the PSPACE-hardness of the problem.
2. For any temporal formula, we effectively build a Büchi automaton that accepts exactly its temporal equilibrium models which allows to provide an automata-based decision procedure. We are able to show that TEL satisfiability can be solved in EXPSPACE and it is PSPACE-hard. Filling the gap is part of future work. Hence, we provide a symbolic representation for sets of temporal equilibrium models raising from temporal formulae.
3. Consequently, given two theories, we provide a decision procedure to check whether they have the same temporal equilibrium models (that is, *regular* equivalence, as opposed to *strong* equivalence).
4. Our proof technique can indeed be adapted to any extension of LTL provided that formulae can be translated into Büchi automata (as happens with LTL with past or LTL with fixed-points operators).

2 Temporal Equilibrium Logic

Let AT $= \{p, q, \ldots\}$ be a countably infinite set of *atoms*. A *temporal formula* is defined with the formal grammar below:

$$\varphi ::= p \mid \bot \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \rightarrow \varphi_2 \mid \mathsf{X}\varphi \mid \varphi_1 \mathsf{U}\, \varphi_2 \mid \varphi_1 \mathsf{R}\, \varphi_2$$

where $p \in$ AT. We will use the standard abbreviations:

$$\neg\varphi \stackrel{\text{def}}{=} \varphi \rightarrow \bot \qquad \top \stackrel{\text{def}}{=} \neg\bot \qquad \varphi \leftrightarrow \varphi' \stackrel{\text{def}}{=} (\varphi \rightarrow \varphi') \wedge (\varphi' \rightarrow \varphi)$$

$$\mathsf{G}\varphi \stackrel{\text{def}}{=} \bot\ \mathsf{R}\, \varphi \qquad \mathsf{F}\varphi \stackrel{\text{def}}{=} \top\ \mathsf{U}\, \varphi$$

The temporal connectives X, G, F, U and R have their standard meaning from LTL. A *theory* Γ is defined as a finite set of temporal formulae.

In the non-temporal case, Equilibrium Logic is defined by introducing a criterion for selecting models based on a non-classical monotonic formalism called the logic of *Here-and-There* (HT) [11], an intermediate logic between intuitionistic and classical propositional calculus. Similarly, TEL will be defined by first introducing a monotonic, intermediate version of LTL, we call the logic of *Temporal Here-and-There* (THT), and then defining a criterion for selecting models in order to obtain nonmonotonicity.

In this way, we will deal with two classes of models. An *LTL model* $\mathbf{H}$ is a map $\mathbf{H} : \mathbb{N} \rightarrow \mathcal{P}(\text{AT})$, viewed as an ω-sequence of propositional valuations. By contrast, the semantics of THT is defined in terms of sequences of pairs of propositional valuations, which can be also viewed as a pair of LTL models. A *THT model* is a pair $\mathbf{M} = (\mathbf{H}, \mathbf{T})$ where $\mathbf{H}$ and $\mathbf{T}$ are LTL models and for

$i \geq 0$, we impose that $\mathbf{H}(i) \subseteq \mathbf{T}(i)$. $\mathbf{H}(i)$ and $\mathbf{T}(i)$ are sets of atoms standing for *here* and *there* respectively. A THT model $\mathbf{M} = (\mathbf{H}, \mathbf{T})$ is said to be *total* when $\mathbf{H} = \mathbf{T}$. The satisfaction relation $\models$ is interpreted as follows on THT models ($\mathbf{M}$ is a THT model and $k \in \mathbb{N}$):

1. $\mathbf{M}, k \models p \overset{\text{def}}{\Leftrightarrow} p \in \mathbf{H}(k)$.
2. $\mathbf{M}, k \models \varphi \wedge \varphi' \overset{\text{def}}{\Leftrightarrow} \mathbf{M}, k \models \varphi$ and $\mathbf{M}, k \models \varphi'$.
3. $\mathbf{M}, k \models \varphi \vee \varphi' \overset{\text{def}}{\Leftrightarrow} \mathbf{M}, k \models \varphi$ or $\mathbf{M}, k \models \varphi'$.
4. $\mathbf{M}, k \models \varphi \rightarrow \varphi' \overset{\text{def}}{\Leftrightarrow}$ for all $\mathbf{H}' \in \{\mathbf{H}, \mathbf{T}\}$, $(\mathbf{H}', \mathbf{T}), k \not\models \varphi$ or $(\mathbf{H}', \mathbf{T}), k \models \varphi'$.
5. $\mathbf{M}, k \models \mathsf{X}\varphi \overset{\text{def}}{\Leftrightarrow} \mathbf{M}, k+1 \models \varphi$.
6. $\mathbf{M}, k \models \varphi \mathsf{U} \varphi' \overset{\text{def}}{\Leftrightarrow}$ there is $j \geq k$ such that $\mathbf{M}, j \models \varphi'$ and for all $j' \in [k, j-1]$, $\mathbf{M}, j' \models \varphi$.
7. $\mathbf{M}, k \models \varphi \mathsf{R} \varphi' \overset{\text{def}}{\Leftrightarrow}$ for all $j \geq k$ such that $\mathbf{M}, j \not\models \varphi'$, there exists $j' \in [k, j-1]$, $\mathbf{M}, j' \models \varphi$.
8. never $\mathbf{M}, k \models \perp$.

A *model* for a theory Γ is a THT model $\mathbf{M}$ such that for every formula $\varphi \in \Gamma$, we have $\mathbf{M}, 0 \models \varphi$. A formula φ is *THT valid* $\overset{\text{def}}{\Leftrightarrow}$ $\mathbf{M}, 0 \models \varphi$ for every THT model $\mathbf{M}$. Similarly, a formula φ is *THT satisfiable* $\overset{\text{def}}{\Leftrightarrow}$ there is a THT model $\mathbf{M}$ such that $\mathbf{M}, 0 \models \varphi$.

As we can see, the main difference with respect to LTL is the interpretation of implication (item 4), that must be checked in both components, $\mathbf{H}$ and $\mathbf{T}$, of $\mathbf{M}$. In fact, it is easy to see that when we take total models $\mathbf{M} = (\mathbf{T}, \mathbf{T})$, THT satisfaction $(\mathbf{T}, \mathbf{T}), k \models \varphi$ collapses to standard LTL satisfaction $\mathbf{T}, k \models \varphi$. We write $\mathbf{T}, k \models \varphi$ instead of $(\mathbf{T}, \mathbf{T}), k \models \varphi$ whenever convenient. For instance, item 4 in the above definition can be rewritten as:

4′. $\mathbf{M}, k \models \varphi \rightarrow \varphi' \overset{\text{def}}{\Leftrightarrow}$ ($\mathbf{M}, k \models \varphi$ implies $\mathbf{M}, k \models \varphi'$) and $\mathbf{T}, k \models \varphi \rightarrow \varphi'$ (LTL satisfaction)

Note that $\mathbf{M}, k \models \neg p$ iff $\mathbf{M}, k \models p \rightarrow \perp$ iff ($\mathbf{M}, k \models p$ implies $\mathbf{M}, k \models \perp$ and $\mathbf{T}, k \models p$ implies $\mathbf{T}, k \models \perp$) iff ($p \notin \mathbf{H}(k)$ and $p \notin \mathbf{T}(k)$).

Similarly, a formula φ is *LTL valid* $\overset{\text{def}}{\Leftrightarrow}$ $\mathbf{M}, 0 \models \varphi$ for every total THT model $\mathbf{M}$ whereas a formula φ is *LTL satisfiable* $\overset{\text{def}}{\Leftrightarrow}$ there is a total THT model $\mathbf{M}$ such that $\mathbf{M}, 0 \models \varphi$. We write $\mathrm{Mod}(\varphi)$ to denote the set of LTL models for φ (restricted to the set of atoms occurring φ denoted by $\mathrm{AT}(\varphi)$).

Obviously, any THT valid formula is also LTL valid, but not the other way around. For instance, the following are THT valid equivalences:

$$\neg(\varphi \wedge \psi) \leftrightarrow \neg\varphi \vee \neg\psi \qquad \mathsf{X}(\varphi \oplus \psi) \leftrightarrow \mathsf{X}\varphi \oplus \mathsf{X}\psi$$
$$\neg(\varphi \vee \psi) \leftrightarrow \neg\varphi \wedge \neg\psi \qquad \mathsf{X} \otimes \varphi \leftrightarrow \otimes \mathsf{X}\varphi$$

for any binary connective $\oplus$ and any unary connective $\otimes$. This means that De Morgan laws are valid, and that we can always shift the X operator to all the operands of any connective. On the contrary, the LTL valid formula $\varphi \vee \neg\varphi$ (known as *excluded middle* axiom) is not THT valid. This is inherited from the intermediate/intuitionistic nature of THT: in fact, the addition of this axiom makes THT collapse into LTL. By adding a copy of this axiom for any atom at any position of the models, we can force that THT models of any formula are total, as stated next.

Proposition 1. *Given a temporal formula φ built over the propositional atoms in* $\mathrm{AT}(\varphi)$*, for every THT model* $(\mathbf{H}, \mathbf{T})$*, the propositions below are equivalent:*

(I) $(\mathbf{H}, \mathbf{T}), 0 \models \varphi \wedge \bigwedge_{p \in \mathrm{AT}(\varphi)} \mathsf{G}(p \vee \neg p)$,

(II) $\mathbf{T}, 0 \models \varphi$ *in LTL, and for* $i \geq 0$ *and* $p \in \mathrm{AT}(\varphi)$*, we have* $p \in \mathbf{H}(i)$ *iff* $p \in \mathbf{T}(i)$.

As a consequence, we can easily encode LTL in THT, since LTL models of φ coincide with its total THT models. Let us state another property whose proof can be obtained by structural induction.

Proposition 2 (Persistence). *For any formula φ, any THT model* $\mathbf{M} = (\mathbf{H}, \mathbf{T})$ *and any* $i \geq 0$*, if* $\mathbf{M}, i \models \varphi$*, then* $\mathbf{T}, i \models \varphi$.

Corollary 1. $(\mathbf{H}, \mathbf{T}), i \models \neg\varphi$ *iff* $\mathbf{T}, i \not\models \varphi$ *in LTL.*

We proceed now to define an ordering relation among THT models, so that only the minimal ones will be *selected* for a temporal theory. Given two LTL models $\mathbf{H}$ and $\mathbf{H}'$, we say that $\mathbf{H}$ is *less than or equal to* $\mathbf{H}'$ (in symbols $\mathbf{H} \leq \mathbf{H}'$) $\stackrel{\text{def}}{\Leftrightarrow}$ for $k \geq 0$, we have $\mathbf{H}(k) \subseteq \mathbf{H}'(k)$. We write $\mathbf{H} < \mathbf{H}'$ if $\mathbf{H} \leq \mathbf{H}'$ and $\mathbf{H} \neq \mathbf{H}'$. The relations $\leq$ and $<$ can be lifted at the level of THT models. Given two THT models $\mathbf{M} = (\mathbf{H}, \mathbf{T})$ and $\mathbf{M}' = (\mathbf{H}', \mathbf{T}')$, $\mathbf{M} \leq \mathbf{M}'$ $\stackrel{\text{def}}{\Leftrightarrow}$ $\mathbf{H} \leq \mathbf{H}'$ and $\mathbf{T} = \mathbf{T}'$. Similarly, we write $\mathbf{M} < \mathbf{M}'$ if $\mathbf{M} \leq \mathbf{M}'$ and $\mathbf{M} \neq \mathbf{M}'$.

Definition 1 (Temporal Equilibrium Model). *A THT model* $\mathbf{M}$ *is a* temporal equilibrium model *(or* TEL model*, for short) of a theory Γ if* $\mathbf{M}$ *is a total model of Γ and there is no* $\mathbf{M}' < \mathbf{M}$ *such that* $\mathbf{M}', 0 \models \Gamma$.

Temporal Equilibrium Logic (TEL) is the logic induced by temporal equilibrium models and it is worth noting that any temporal equilibrium model of Γ is a *total* THT model of the form $(\mathbf{T}, \mathbf{T})$ (by definition). The corresponding LTL model $\mathbf{T}$ of Γ is said to be a *temporal stable model* of Γ.

When we restrict the syntax to non-modal theories and semantics to HT interpretations $\langle \mathbf{H}(0), \mathbf{T}(0) \rangle$ we talk about (non-temporal) equilibrium models, which coincide with stable models in their most general definition [8].

The TEL satisfiability problem consists in determining whether a temporal formula has a TEL model. As an example, consider the formula

$$\mathsf{G}(\neg p \rightarrow \mathsf{X} p) \tag{1}$$

Its intuitive meaning corresponds to the logic program consisting of rules of the form: $p(s(X)) \leftarrow \mathit{not}\ p(X)$ where time has been reified as an extra parameter $X = 0, s(0), s(s(0)), \ldots$. Notice that the interpretation of $\neg$ is that of default negation *not* in logic programming. In this way, (1) is saying that, at any situation $i \geq 0$, if there is no evidence on p, then p will become true in the next state $i+1$. In the initial state, we have no evidence on p, so this will imply $\mathsf{X}p$. As a result $\mathsf{XX}p$ will have no applicable rule and thus will be false by default, and so on. It is easy to see that the unique temporal stable model of (1) is defined by the formula $\neg p \wedge \mathsf{G}(\neg p \leftrightarrow \mathsf{X}p)$.

It is worth noting that an LTL satisfiable formula may have no temporal stable model. As a simple example (well-known from non-temporal ASP) the logic program rule $\neg p \rightarrow p$, whose only (classical) model is $\{p\}$, has no stable models. This is because if we take a model $\mathbf{M} = (\mathbf{H}, \mathbf{T})$ where p holds in $\mathbf{T}$, then (Corollary 1) $\mathbf{M} \not\models \neg p$ and so $\mathbf{M} \models \neg p \rightarrow p$ true regardless $\mathbf{H}$, so we can take a strictly smaller $\mathbf{H} < \mathbf{T}$ whose only difference with respect to $\mathbf{T}$ is that p does not hold. On the other hand, if we take any $\mathbf{T}$ in which p does not hold, then $\mathbf{M} \models \neg p$ and so $\neg p \rightarrow p$ would make p true both in $\mathbf{H}$ and $\mathbf{T}$ reaching a contradiction. When dealing with logic programs, it is well-known that non-existence of stable models is always due to a kind of cyclic dependence on default negation like this.

In the temporal case, however, non-existence of temporal stable models may also be due to a lack of a finite justification for satisfying the criterion of minimal knowledge. As an example, take the formula $\mathsf{GF}p$, typically used in LTL to assert that property p occurs infinitely often. This formula has no temporal stable models: all models must contain infinite occurrences of p and there is no way to establish a minimal $\mathbf{H}$ among them. Thus, formula $\mathsf{GF}p$ is LTL satisfiable but it has no temporal stable model. By contrast, forthcoming Proposition 4 states that for a large class of temporal formulae, LTL satisfiability is equivalent to THT satisfiability and TEL satisfiability.

3 Automata-Based Approach for LTL in a Nutshell

Before presenting our decision procedures, let us briefly recall what are the main ingredients of the automata-based approach. It consists in reducing logical problems into automata-based decision problems in order to take advantage of known results from automata theory. The most standard target problems on automata used in this approach are the nonemptiness problem (checking whether an automaton admits at least one accepting computation) and the inclusion problem (checking whether the language accepted by the automaton $\mathcal{A}$ is included in the language accepted by the automaton $\mathcal{B}$). In a pioneering work [4] Büchi introduced a class of automata showing that they are equivalent to formulae in monadic second-order logic (MSO) over $(\mathbb{N}, <)$.

In full generality, here are a few desirable properties of the approach. The reduction should be conceptually simple, see the translation from LTL formulae into alternating automata [27]. Formula structure is reflected directly in the transition formulae of alternating automata. The computational complexity of the automata-based target problem should be well-characterized – see, for instance, the translation from PDL formulae into nondeterministic Büchi tree automata [28]. It is also highly desirable that not only the reduction is conceptually simple but also that it is semantically faithful so that the automata involve in the target instance are closely related to the instance of the original logical problem. Last but not least, preferrably, the reduction might allow to obtain the optimal complexity for the source logical problem.

3.1 Basics on Büchi automata

We recall that a *Büchi automaton* $\mathcal{A}$ is a tuple $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$ such that Σ is a finite *alphabet*, Q is a finite set of *states*, $Q_0 \subseteq Q$ is the set of *initial* states, the *transition relation* δ is a subset of $Q \times \Sigma \times Q$ and $F \subseteq Q$ is a set of *final* states. Given $q \in Q$ and $a \in \Sigma$, we also write $\delta(q, a)$ to denote the set of states q' such that $(q, a, q') \in \delta$.

A *run* ρ of $\mathcal{A}$ is a sequence $q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \ldots$ such that for every $i \geq 0$, $(q_i, a_i, q_{i+1}) \in \delta$ (also written $q_i \xrightarrow{a_i} q_{i+1}$). The run ρ is *accepting* if $q_0 \in Q_0$ is initial and some state of F is repeated infinitely often in ρ: $\inf(\rho) \cap F \neq \emptyset$ where we let $\inf(\rho) = \{q \in Q : \forall\, i,\ \exists\, j > i,\ q = q_j\}$. The *label* of ρ is the word $\sigma = a_0 a_1 \cdots \in \Sigma^\omega$. The automaton $\mathcal{A}$ *accepts* the language $\mathrm{L}(\mathcal{A})$ of ω-words $\sigma \in \Sigma^\omega$ such that there exists an accepting run of $\mathcal{A}$ on the word σ, i.e., with label σ.

Now, we introduce a standard generalization of the Büchi acceptance condition by considering conjunctions of classical Büchi conditions. A *generalized Büchi automaton* (GBA) is a structure $\mathcal{A} = (\Sigma, Q, Q_0, \delta, \{F_1, \ldots, F_k\})$ such that $F_1, \ldots, F_k \subseteq Q$ and Σ, Q, Q_0 and δ are defined as for Büchi automata. A run is defined as for Büchi automata and a run ρ of $\mathcal{A}$ is *accepting* iff the first state is initial and for $i \in [1, n]$, we have $\inf(\rho) \cap F_i \neq \emptyset$. It is known that every GBA $\mathcal{A}$ can be easily translated in logarithmic space into a Büchi automaton, preserving the language of accepted ω-words (see e.g. [21]). Moreover, the nonemptiness problem for GBA or BA is known to be NLogSpace-complete.

3.2 From LTL formulae to Büchi automata

We recall below how to define a Büchi automaton that accepts the linear models of an LTL formula. Given an LTL formula φ, we define its *closure* $cl(\varphi)$ to denote a finite set of formulae that are relevant to check the satisfiability of φ. For each LTL formula φ, we define its *main components* (if any) according to the table below:

formula φ	main components
p or $\neg p$	none
$\neg\neg\psi$ or $\mathsf{X}\psi$	ψ
$\neg\mathsf{X}\psi$	$\neg\psi$
$\psi_1 \mathsf{U}\, \psi_2$ or $\psi_1 \wedge \psi_2$ or $\psi_1 \mathsf{R}\, \psi_2$	ψ_1, ψ_2
$\neg(\psi_1 \mathsf{U}\, \psi_2)$ or $\neg(\psi_1 \wedge \psi_2)$ or $\neg(\psi_1 \mathsf{R}\, \psi_2)$	$\neg\psi_1$, $\neg\psi_2$

We write $cl(\varphi)$ to denote the least set of formulae such that $\varphi \in cl(\varphi)$ and $cl(\varphi)$ is closed under main components. It is routine to check that $\mathrm{card}(cl(\varphi)) \leq$

$|\varphi|$ (the size of φ). Moreover, one can observe that if $\psi \in cl(\varphi)$, then for each immediate subformula (if any), either it belongs to $cl(\varphi)$ or its negation belongs to $cl(\varphi)$. A subset $\Gamma \subseteq cl(\varphi)$ is *consistent and fully expanded* whenever

- $\psi_1 \wedge \psi_2 \in \Gamma$ implies $\psi_1, \psi_2 \in \Gamma$,
- $\neg(\psi_1 \wedge \psi_2) \in \Gamma$ implies $\neg\psi_1 \in \Gamma$ or $\neg\psi_2 \in \Gamma$,
- $\neg\neg\psi \in \Gamma$ implies $\psi \in \Gamma$,
- Γ does not contain a contradictory pair ψ and $\neg\psi$.

The pair of consistent and fully expanded sets (Γ_1, Γ_2) is *one-step consistent* $\stackrel{\text{def}}{\Leftrightarrow}$

1. $\mathsf{X}\psi \in \Gamma_1$ implies $\psi \in \Gamma_2$ and $\neg\mathsf{X}\psi \in \Gamma_1$ implies $\neg\psi \in \Gamma_2$.
2. $\psi_1 \mathsf{U}\, \psi_2 \in \Gamma_1$ implies $\psi_2 \in \Gamma_1$ or ($\psi_1 \in \Gamma_1$ and $\psi_1 \mathsf{U}\, \psi_2 \in \Gamma_2$),
3. $\neg(\psi_1 \mathsf{U}\, \psi_2) \in \Gamma_1$ implies $\neg\psi_2 \in \Gamma_1$ and ($\neg\psi_1 \in \Gamma_1$ or $\neg(\psi_1 \mathsf{U}\, \psi_2) \in \Gamma_2$).
4. $\psi_1 \mathsf{R}\, \psi_2 \in \Gamma_1$ implies $\psi_2 \in \Gamma_1$ and ($\psi_1 \in \Gamma_1$ or $\psi_1 \mathsf{R}\, \psi_2 \in \Gamma_2$),
5. $\neg(\psi_1 \mathsf{R}\, \psi_2) \in \Gamma_1$ implies $\neg\psi_2 \in \Gamma_1$ or ($\neg\psi_1 \in \Gamma_1$ and $\neg(\psi_1 \mathsf{R}\, \psi_2) \in \Gamma_2$).

Given an LTL formula φ, let us build the generalized Büchi automaton $\mathcal{A}_\varphi = (\Sigma, Q, Q_0, \delta, \{F_1, \ldots, F_k\})$ where

- $\Sigma = \mathcal{P}(\mathrm{AT}(\varphi))$ and Q is the set of consistent and fully expanded sets.
- $Q_0 = \{\Gamma \in Q : \varphi \in \Gamma\}$.
- $\Gamma \xrightarrow{a} \Gamma' \in \delta \stackrel{\text{def}}{\Leftrightarrow} (\Gamma, \Gamma')$ is one-step consistent, $(\Gamma \cap \mathrm{AT}(\varphi)) \subseteq a$ and $\{p : \neg p \in \Gamma\} \cap a = \emptyset$.
- If the temporal operator U does not occur in φ, then $F_1 = Q$ and $k = 1$. Otherwise, suppose that $\{\psi_1 \mathsf{U}\, \psi'_1, \ldots, \psi_k \mathsf{U}\, \psi'_k\}$ is the set of U-formulae from $cl(\varphi)$. Then, for every $i \in [1, k]$, $F_i = \{\Gamma \in Q : \psi_i \mathsf{U}\, \psi'_i \notin \Gamma$ or $\psi'_i \in \Gamma\}$.

It is worth observing that $\mathrm{card}(Q) \leq 2^{|\varphi|}$ and $\mathcal{A}_\varphi$ can be built in exponential time in $|\varphi|$.

Proposition 3. *[29]* $\mathrm{Mod}(\varphi) = \mathrm{L}(\mathcal{A}_\varphi)$, *i.e. for every* $a_0 a_1 \cdots \in \Sigma^\omega$, $a_0 a_1 \cdots \in \mathrm{L}(\mathcal{A}_\varphi)$ *iff* $\mathbf{T}, 0 \models \varphi$ *where for all* $i \in \mathbb{N}$, $\mathbf{T}(i) = a_i$.

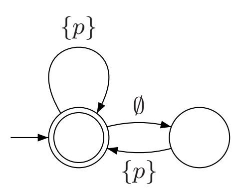

Fig. 1. Büchi automaton for models of $\mathsf{G}(\neg p \rightarrow \mathsf{X}p)$ (over the alphabet $\{\emptyset, \{p\}\}$)

Figure 1 presents a Büchi automaton recognizing the models for $\mathsf{G}(\neg p \rightarrow \mathsf{X}p)$ over the alphabet $\{\emptyset, \{p\}\}$. The automaton obtained from the above systematic construction would be a bit larger since $cl(\mathsf{G}(\neg p \rightarrow \mathsf{X}p))$ has about 2^4 subsets. However, the systematic construction has the advantage to be generic. Other translations exist with other advantages, for instance to build small automata, see e.g. [7]. However, herein, we need to use the following properties (apart from the correctness of the reduction): (1) the size of each state of $\mathcal{A}_\varphi$ is linear in the size of φ, (2) it can be checked if a state is initial [resp. final] in linear space in the size of φ and (3) given two subsets X, X' of $cl(\varphi)$ and $a \in \Sigma$, one can check in linear space in the size of φ whether $X \xrightarrow{a} X'$ is a transition of $\mathcal{A}_\varphi$ (each transition of $\mathcal{A}_\varphi$ can be checked in linear space in the size of φ).

These are key points towards the PSPACE upper bound for LTL satisfiability since the properties above are sufficient to check the nonemptiness of $\mathcal{A}_\varphi$ in nondeterministic polynomial space in the size of φ (guess on-the-fly a prefix and a loop of length at most exponential) and then invoke Savitch Theorem [24] to eliminate nondeterminism. We will use similar arguments to establish that TEL satisfiability can be solved in EXPSPACE.

4 Building TEL Models with Büchi Automata

In this section, we provide an automata-based approach to determine whether a formula φ built over the atoms $\{p_1, \ldots, p_n\}$ has a TEL model. This is the place where starts our main contribution. To do so, we build a Büchi automaton $\mathcal{B}$ over the alphabet $\Sigma = \mathcal{P}(\{p_1, \ldots, p_n\})$ such that $\mathrm{L}(\mathcal{B})$ is equal to the set of TEL models for φ. Moreover, nonemptiness can be checked in EXPSPACE, which allows to answer the open problem about the complexity of determining whether a temporal formula has a TEL model.

Each model $\mathbf{M} = \langle \mathbf{H}, \mathbf{T} \rangle$ restricted to the atoms in $\{p_1, \ldots, p_n\}$ can be encoded into an LTL model $\mathbf{H}'$ over the alphabet $\Sigma' = \mathcal{P}(\{p_1, \ldots, p_n, p'_1, \ldots, p'_n\})$ such that for $i \geq 0$, $\mathbf{H}'(i) = (\mathbf{T}(i) \cap \{p_1, \ldots, p_n\}) \cup \{p'_j : p_j \in \mathbf{H}(i), j \in [1, n]\}$. In that case, we write $\mathbf{H}' \approx \mathbf{M}$.

Lemma 1.

- **(I)** *For every THT model* $\mathbf{M} = \langle \mathbf{H}, \mathbf{T} \rangle$ *restricted to atoms in* $\{p_1, \ldots, p_n\}$*, there is a unique LTL model* $\mathbf{H}'$ *such that* $\mathbf{H}' \approx \mathbf{M}$.
- **(II)** *For every LTL model* $\mathbf{H}' : \mathbb{N} \to \Sigma'$ *such that* $\mathbf{H}', 0 \models \bigwedge_{i \in [1,n]} \mathsf{G}(p'_i \to p_i)$*, there is a unique THT model* $\mathbf{M} = \langle \mathbf{H}, \mathbf{T} \rangle$ *restricted to atoms s.t.* $\mathbf{H}' \approx \mathbf{M}$.

In (II), a THT model $\mathbf{M}$ can be built thanks to the satisfaction of the formula $\bigwedge_{i \in [1,n]} \mathsf{G}(p'_i \to p_i)$ by $\mathbf{H}'$, which guarantees that for all $i \in \mathbb{N}$, we have $\mathbf{H}(i) \subseteq \mathbf{T}(i)$. The proof is by an easy verification. This guarantees a clear isomorphism between two sets of models. In order to complete this model-theoretical correspondence, let us define the translation f between temporal formulae:

- f is homomorphic for conjunction, disjunction and temporal operators,
- $f(\perp) \stackrel{\text{def}}{=} \perp$, $f(p_i) \stackrel{\text{def}}{=} p'_i$ and $f(\psi \to \psi') \stackrel{\text{def}}{=} (\psi \to \psi') \wedge (f(\psi) \to f(\psi'))$.

Lemma 2. *Let* φ *be a temporal formula built over the atoms in* $\{p_1, \ldots, p_n\}$ *and* $\mathbf{M}$ *restricted to* $\{p_1, \ldots, p_n\}$ *and* $\mathbf{H}'$ *be models such that* $\mathbf{H}' \approx \mathbf{M}$*. For* $l \geq 0$*, we have* $\mathbf{H}', l \models f(\psi)$ *iff* $\mathbf{M}, l \models \psi$ *for every subformula* ψ *of* φ.

The proof is by an easy structural induction. So, there is a polynomial-time reduction from THT satisfiability into LTL satisfiability by considering the mapping $f(\cdot) \wedge \bigwedge_{i \in [1,n]} \mathsf{G}(p'_i \rightarrow p_i)$.

Let $\mathcal{A}_1$ be the Büchi automaton such that $\mathrm{L}(\mathcal{A}_1) = \mathrm{Mod}(\varphi)$, following any construction similar to [29] (see Section 3.2). The set $\mathrm{L}(\mathcal{A}_1)$ can be viewed as the set of total THT models of φ. Let φ' be the formula $f(\varphi) \wedge \bigwedge_{i \in [1,n]} \mathsf{G}(p'_i \rightarrow p_i)$.

Lemma 3. *The set of LTL models for the formula φ' corresponds to the set of THT models for the temporal formula φ.*

For instance, taking the formula $\varphi = \mathsf{G}(\neg p \rightarrow \mathsf{X}p)$, we can compute its THT models $\mathbf{M}$ by obtaining the corresponding LTL models (with atoms p and p') for the formula below:

$$\begin{aligned} \varphi' &= f(\ \mathsf{G}(\neg p \rightarrow \mathsf{X}p)\ ) \ \wedge\ \mathsf{G}(p' \rightarrow p) \\ &= \mathsf{G}\big(\ (\neg p \rightarrow \mathsf{X}p) \wedge (\neg p \wedge \neg p' \rightarrow \mathsf{X}p')\ \big) \ \wedge\ \mathsf{G}(p' \rightarrow p) \end{aligned}$$

Figure 2 presents a Büchi automaton for the models of the formula $f(\mathsf{G}(\neg p \rightarrow \mathsf{X}p)) \wedge \mathsf{G}(p' \rightarrow p)$ over the alphabet $\{\emptyset, \{p\}, \{p'\}, \{p, p'\}\}$. Hence, we provide a symbolic representation for the THT models of $\mathsf{G}(\neg p \rightarrow \mathsf{X}p)$. For instance, reading the letter $\{p\}$ at position i corresponds to a pair $(\mathbf{H}(i), \mathbf{T}(i))$ with $p \notin \mathbf{H}(i)$ and $p \in \mathbf{T}(i)$. Similarly, reading the letter $\{p, p'\}$ at position i corresponds to a pair $(\mathbf{H}(i), \mathbf{T}(i))$ with $p \in \mathbf{H}(i)$ and $p \in \mathbf{T}(i)$. However, $\{p'\}$ cannot be read since $\mathbf{H}(i) \subseteq \mathbf{T}(i)$.

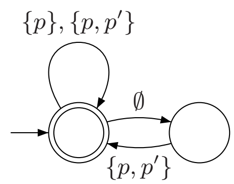

Fig. 2. Büchi automaton for models of $f(\mathsf{G}(\neg p \rightarrow \mathsf{X}p)) \wedge \mathsf{G}(p' \rightarrow p)$

Hence, φ is THT satisfiable iff φ' is LTL satisfiable.

The map f shall be also useful to show Proposition 4 below, becoming a key step to obtain PSPACE-hardness results (see e.g. Theorem 2). Proposition 4 below states that for a large class of formulae, LTL satisfiability is equivalent to TEL satisfiability.

Proposition 4. *Let φ be temporal formula built over the connectives $\vee$, $\wedge$, $\rightarrow$, X and U and such that $\rightarrow$ occurs only in subformulae of the form $p \rightarrow \perp$ with $p \in \mathrm{AT}$. The propositions below are equivalent: (I) φ is LTL satisfiable; (II) φ is THT satisfiable; (III) φ has a temporal stable model, i.e. φ is TEL satisfiable.*

Proof. Let $\mathrm{L}^{+}(\vee, \wedge, \mathsf{X}, \mathsf{U})$ be the class of temporal formulae involved in the statement. Every temporal formula φ in $\mathrm{L}^{+}(\vee, \wedge, \mathsf{X}, \mathsf{U})$ states a *guarantee property* in the sense of [30] (see also [14]). In particular, this means that

(P1) if $\mathbf{T}, 0 \models \varphi$ in the LTL sense, then there is $N \geq 0$ such that for any $\mathbf{T}'$ that only agrees with $\mathbf{T}$ on the positions in $[0, N]$, we also have $\mathbf{T}', 0 \models \varphi$ (φ states a guarantee property).

In particular, we can impose that for $l > N$, we have $\mathbf{T}'(l) = \emptyset$. Moreover, since the set of sequences of length $N+1$ indexed by subsets of atoms occurring in φ is finite, if φ is LTL satisfiable, then there is an LTL model $\mathbf{T}$ such that $\mathbf{T}, 0 \models \varphi$ and no $\mathbf{T}' < \mathbf{T}$ verifies $\mathbf{T}', 0 \models \varphi$ (this property does not hold for every LTL formulae, consider $\mathsf{GF}p$). (II) implies (I) is by Proposition 2 and (III) implies (II) is by definition of temporal stable model whereas (III) implies (I) is by definition of LTL satisfiability. It remains to show that (I) implies (III).

Suppose that φ is LTL satisfiable. From the previous properties, we have seen that there is a minimal LTL model $\mathbf{T}$ such that $\mathbf{T}, 0 \models \varphi$. The argument is by *reductio ad absurdum*. Suppose that there is $\mathbf{H} < \mathbf{T}$ such that $(\mathbf{H}, \mathbf{T}), 0 \models \varphi$. By minimality of $\mathbf{T}$, we have $\mathbf{H}, 0 \not\models \varphi$. Since $\mathbf{T}$ is minimal and has an infinite suffix of the form $\emptyset^\omega$, there is a finite amount of positions $l_1, \ldots, l_N$ such that $\mathbf{H}(l_i) \subset \mathbf{T}(l_i)$ (strict inclusion).

Let $\mathbf{H}'$ be the LTL model such that $\mathbf{H}' \approx (\mathbf{H}, \mathbf{T})$. We have seen that $\mathbf{H}', 0 \models f(\varphi)$ (Lemma 2). Let us define the map f', that we shall apply to φ, as a slight variant of f:

- f' is homomorphic for conjunction, disjunction and temporal operators,
- $f'(p_i) \stackrel{\text{def}}{=} (p'_i \wedge p_i)$ and $f'(\neg p_i) \stackrel{\text{def}}{=} (\neg p_i \wedge \neg p'_i)$.

Since, $\mathbf{H}', 0 \models \bigwedge_{i \in [1,n]} \mathsf{G}(p'_i \rightarrow p_i)$, we also have $\mathbf{H}', 0 \models f'(\varphi)$. Let $\mathbf{H}''$ be the variant of $\mathbf{H}'$ such that for $l \not\in \{l_1, \ldots, l_N\}$, $\mathbf{H}''(l) \stackrel{\text{def}}{=} \mathbf{H}'(l)$ and for $i \in [1, N]$ and $p \in \text{AT}$, if $p \not\in \mathbf{H}(l_i)$ and $p \in \mathbf{T}(l_i)$, then $p, p' \not\in \mathbf{H}''(l_i)$, otherwise ($p, p' \in \mathbf{H}''(l_i) \stackrel{\text{def}}{\Leftrightarrow} p \in \mathbf{T}(l_i)$). Observe that in $\mathbf{H}''$ the valuations on $\{p_1, \ldots, p_n\}$ and $\{p'_1, \ldots, p'_n\}$ agree and the 'atomic' formulae in $f'(\varphi)$ are the form either $(p'_i \wedge p_i)$ or $(\neg p'_i \wedge \neg p_i)$. Consequently, $\mathbf{H}'', 0 \models f'(\varphi)$ and by Lemma 2, $\mathbf{T}', 0 \models \varphi$ where $\mathbf{T}'(l) = \mathbf{H}''(l) \cap \{p_1, \ldots, p_n\}$ for $l \geq 0$. By construction of $\mathbf{T}'$, we have $\mathbf{T}' < \mathbf{T}$, which leads to a contradiction. Hence, $(\mathbf{T}, \mathbf{T})$ is a temporal stable model for φ. □

Corollary 2. *THT satisfiability problem is* PSpace*-complete.*

Proof. The translation f requires only polynomial-time and since LTL satisfiability is PSpace-complete [25], we get that THT satisfiability is in PSpace. It remains to show the PSpace lower bound.

To do so, we can just observe that, as proved by Proposition 1, LTL satisfiability (which is PSpace-complete) can be encoded into THT satisfiability using the translation from Proposition 1, which can be performed in linear time. Indeed, it just adds a formula $\mathsf{G}(p \vee \neg p)$ per each atom $p \in \text{AT}$. □

We can strengthen the mapping φ' to obtain not only THT models of φ but also to constrain them to be strictly non-total (that is $\mathbf{H} < \mathbf{T}$) as follows

$$\varphi'' \stackrel{\text{def}}{=} \varphi' \wedge \bigvee_{i \in [1,n]} \mathsf{F}((p'_i \rightarrow \bot) \wedge p_i)$$

φ'' characterizes the non-total THT models of the formula φ. The generalized disjunction ensures that at some position j, $\mathbf{H}(j) \subset \mathbf{T}(j)$ (strict inclusion).

Lemma 4. *The set of LTL models for the formula φ'' corresponds to the set of non-total THT models for the temporal formula φ.*

The proof is again by structural induction. Let $\mathcal{A}_2$ be the Büchi automaton such that $\mathrm{L}(\mathcal{A}_2) = \mathrm{Mod}(\varphi'')$, following again any construction similar to [29] (see Section 3.2). $\mathrm{L}(\mathcal{A}_2)$ contains exactly the non-total THT models of φ.

Let $h : \Sigma' \rightarrow \Sigma$ be a map (renaming) between the two finite alphabets such that $h(a) = a \cap \{p_1, \ldots, p_n\}$. h consists in erasing the atoms from $\{p'_1, \ldots, p'_n\}$ h can be naturally extended as an homomorphism between finite words, infinite words and as a map between languages. Similary, given a Büchi automaton $\mathcal{A}_2 = (\Sigma', Q, Q_0, \delta, F)$, we write $h(\mathcal{A}_2)$ to denote the Büchi automaton $(\Sigma, Q, Q_0, \delta', F)$ such that $q \xrightarrow{a} q' \in \delta' \stackrel{\text{def}}{\Leftrightarrow}$ there is $b \in \Sigma'$ such that $q \xrightarrow{b} q' \in \delta$ and $h(b) = a$. Obviously, $\mathrm{L}(h(\mathcal{A}_2)) = h(\mathrm{L}(\mathcal{A}_2))$. Indeed, the following propositions imply each other:

1. $a_0 a_1 \cdots \in \mathrm{L}(\mathcal{A}_2)$,
2. $h(a_0)h(a_1) \cdots \in h(\mathrm{L}(\mathcal{A}_2))$ (by definition of h on languages),
3. $h(a_0)h(a_1) \cdots \in \mathrm{L}(h(\mathcal{A}_2))$ (by definition of h on $\mathcal{A}_2$).

The inclusion $\mathrm{L}(h(\mathcal{A}_2)) \subseteq h(\mathrm{L}(\mathcal{A}_2))$ can be shown in a similar way. So, $\mathrm{L}(h(\mathcal{A}_2))$ can be viewed as the set of total THT models for φ having a strictly smaller THT model.

Proposition 5. *φ has a TEL model iff* $\mathrm{L}(\mathcal{A}_1) \cap (\Sigma^\omega \setminus \mathrm{L}(h(\mathcal{A}_2))) \neq \emptyset$.

Proof. A TEL model $\mathbf{M} = (\mathbf{H}, \mathbf{T})$ for φ satisfies the following properties:

1. $\mathbf{M}, 0 \models \varphi$ and $\mathbf{H} = \mathbf{T}$.
2. For no $\mathbf{H}' < \mathbf{H}$, we have $(\mathbf{H}', \mathbf{T}), 0 \models \varphi$.

We have seen that $\mathrm{L}(\mathcal{A}_1)$ contains exactly the LTL models of φ, i.e. the total THT models satisfying φ. For taking care of condition (2.), by construction, $\mathcal{A}_2$ accepts the non-total THT models for φ whereas $\mathrm{L}(h(\mathcal{A}_2))$ contains the total THT models for φ having a strictly smaller THT model satisfying φ, the negation of (2.). Hence, $(\mathbf{T}, \mathbf{T})$ is a TEL model for φ iff $\mathbf{T} \in \mathrm{L}(\mathcal{A}_1)$ and $\mathbf{T} \notin \mathrm{L}(h(\mathcal{A}_2))$. □

Hence, the set of TEL models for a given φ forms an ω-regular language.

Proposition 6. *For each temporal formula φ, one can effectively build a Büchi automaton that accepts exactly the TEL models for φ.*

Proof. The class of languages recognized by Büchi automata (the class of ω-regular languages) is effectively closed under union, intersection and complementation. Moreover, it is obviously closed under the renaming operation. Since $\mathcal{A}_1$, $\mathcal{A}_2$, and $h(\mathcal{A}_2)$ are Büchi automata, one can build a Büchi automaton $\mathcal{A}'$ such that $\mathrm{L}(\mathcal{A}') = \Sigma^\omega \setminus \mathrm{L}(h(\mathcal{A}_2))$. Similarly, one can effectively build a Büchi automaton $\mathcal{B}_\varphi$ such that $\mathrm{L}(\mathcal{B}_\varphi) = \mathrm{L}(\mathcal{A}_1) \cap \mathrm{L}(\mathcal{A}')$. Complementation can be performed using the constructions in [26] or in [23] (if optimality is required). Roughly speaking, complementation induces an exponential blow-up. □

Figure 3 presents a Büchi automaton accepting the (unique) temporal equilibrium model for φ. The next step consists in showing that the nonemptiness check can be done in exponential space.

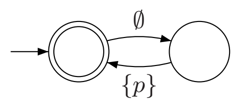

Fig. 3. Büchi automaton for stable models of $\mathsf{G}(\neg p \rightarrow \mathsf{X}p)$

Proposition 7. *Checking whether a TEL formula has a TEL model can be done in* EXPSPACE.

Proof. Let φ be a temporal formula and $\mathcal{A}_1$ and $\mathcal{A}_2$ be Büchi automata such that $\mathrm{L}(\mathcal{A}_1) \cap (\Sigma^\omega \setminus \mathrm{L}(h(\mathcal{A}_2)))$ accepts exactly the TEL models for φ. We shall show that nonemptiness of the language can be tested in exponential space. With the construction of $\mathcal{A}_1$ using [29], we have seen that

1. the size of each state of $\mathcal{A}_1$ is linear in the size of φ (written $|\varphi|$),
2. it can be checked if a state is initial [resp. final] in linear space in $|\varphi|$,
3. each transition of $\mathcal{A}_1$ can be checked in linear space in $|\varphi|$,
4. $\mathcal{A}_1$ has a number of states exponential in $|\varphi|$.

Similarly, let us observe the following simple properties:

(a) φ'' is of linear size in $|\varphi|$; (b) Automaton $\mathcal{A}_2$ can be built from the formula φ'' using the construction in [29]; (c) $\mathcal{A}_2$ and $h(\mathcal{A}_2)$ have the same sets of states, initial states and final states and checking whether a transition belongs to $h(\mathcal{A}_2)$ is not more complex than checking whether a transition belongs to $\mathcal{A}_2$. So,

1. the size of each state of $h(\mathcal{A}_2)$ is linear in $|\varphi|$,
2. it can be checked if a state is initial [resp. final] in linear space in $|\varphi|$,
3. each transition of $h(\mathcal{A}_2)$ can be checked in linear space in $|\varphi|$.
4. $h(\mathcal{A}_2)$ has a number of states exponential in $|\varphi|$.

Using the complementation construction from [26] (the construction in [23] would be also fine) to complement $h(\mathcal{A}_2)$, one can obtain a Büchi automaton $\mathcal{A}'$ such that $\mathrm{L}(\mathcal{A}') = \Sigma^\omega \setminus \mathrm{L}(h(\mathcal{A}_2))$ and

1. the size of each state of $\mathcal{A}'$ is exponential in $|\varphi|$,
2. it can be checked if a state is initial [resp. final] in exponential space in $|\varphi|$,
3. each transition of $\mathcal{A}'$ can be checked in exponential space in $|\varphi|$.
4. $\mathcal{A}'$ has a number of states doubly exponential in $|\varphi|$.

Indeed, $h(\mathcal{A}_2)$ is already of exponential size in $|\varphi|$. So, using the above-mentioned property, one can check on-the-fly whether $\mathrm{L}(\mathcal{A}_1) \cap \mathrm{L}(\mathcal{A}')$ is nonempty by guessing a synchronized run of length at most double exponential (between the automata $\mathcal{A}_1$ and $\mathcal{A}'$) and check that it satisfies the acceptance conditions of both automata. At any stage of the algorithm, at most 2 product states need to be stored and this requires exponential space. Similarly, counting until a double exponential value requires only an exponential amount of bits. Details are omitted but the very algorithm is based on standard arguments for checking on-the-fly graph accessibility and checking nonemptiness of the intersection of two languages accepted by Büchi automata (similar arguments are used in [26, Lemma 2.10]). By Savitch Theorem [24], nondeterminism can be eliminated, providing the promised ExpSpace upper bound. ◻

Theorem 1. *Checking whether a formula has a TEL model is* PSpace*-hard.*

Proof. We can use again the linear encoding in Proposition 1 and observe that any THT model $(\mathbf{T},\mathbf{T})$ of $\psi = \varphi \wedge \bigwedge_{p \in \mathrm{AT}(\varphi)} \mathsf{G}(p \vee \neg p)$ will also be a TEL model of φ, since there are no non-total models for ψ and thus $(\mathbf{T},\mathbf{T})$ will always be minimal. But then $\mathbf{T} \models \varphi$ in LTL iff $(\mathbf{T},\mathbf{T}) \models \psi$ in THT iff $(\mathbf{T},\mathbf{T})$ is a TEL model of ψ. Thus LTL satisfiability can be reduced to TEL satisfiability and so the latter problem is PSpace-hard. ◻

Theorem 2. *Checking whether two temporal formulae have the same TEL models is decidable in* ExpSpace *and it is* PSpace*-hard.*

Proof. Let $\varphi \in \mathrm{L}^+(\vee, \wedge, \mathsf{X}, \mathsf{U})$ and $\psi = \mathsf{GF}p_1$. We recall that ψ has no temporal equilibrium model. The propositions below are equivalent: (a) φ is LTL satisfiable, (b) φ has a temporal equilibrium model and (c) φ and ψ have distinct sets of temporal equilibrium models. Since LTL satisfiability for the fragment $\mathrm{L}^+(\vee, \wedge, \mathsf{X}, \mathsf{U})$ is PSpace-hard, coPSpace= PSpace and ((a) iff (c)), then the equivalence problem with temporal equilibrium models is PSpace-hard.

Let φ and ψ be two temporal formulae built over the same set of atoms and $\mathcal{A}_1^{\varphi}$, $\mathcal{A}_2^{\varphi}$, $\mathcal{A}_1^{\psi}$ and $\mathcal{A}_2^{\psi}$ be Büchi automata such that $\mathrm{L}(\mathcal{A}_1^{\varphi}) \cap (\Sigma^{\omega} \setminus \mathrm{L}(h(\mathcal{A}_2^{\varphi})))$ recognizes the temporal equilibrium models for φ and $\mathrm{L}(\mathcal{A}_1^{\psi}) \cap (\Sigma^{\omega} \setminus \mathrm{L}(h(\mathcal{A}_2^{\psi})))$ recognizes the temporal equilibrium models for ψ. So, φ and ψ have distinct sets of temporal equilibrium models iff one of the sets below is non-empty.

(I) $\mathrm{L}(\mathcal{A}_1^{\varphi}) \cap (\Sigma^{\omega} \setminus \mathrm{L}(h(\mathcal{A}_2^{\varphi}))) \cap (\Sigma^{\omega} \setminus \mathrm{L}(\mathcal{A}_1^{\psi}))$,
(II) $\mathrm{L}(\mathcal{A}_1^{\varphi}) \cap (\Sigma^{\omega} \setminus \mathrm{L}(h(\mathcal{A}_2^{\varphi}))) \cap \mathrm{L}(h(\mathcal{A}_2^{\psi}))$,
(III) $\mathrm{L}(\mathcal{A}_1^{\psi}) \cap (\Sigma^{\omega} \setminus \mathrm{L}(h(\mathcal{A}_2^{\psi}))) \cap (\Sigma^{\omega} \setminus \mathrm{L}(\mathcal{A}_1^{\varphi}))$,
(IV) $\mathrm{L}(\mathcal{A}_1^{\psi}) \cap (\Sigma^{\omega} \setminus \mathrm{L}(h(\mathcal{A}_2^{\psi}))) \cap \mathrm{L}(h(\mathcal{A}_2^{\varphi}))$.

A nondeterministic algorithm running in exponential space is designed as follows:

1. Guess which sets among (I)–(IV) is tested for nonemptiness.
2. Run a nondeterministic algorithm in exponential space by synchronizing the transitions of the three automata (one or two of them are designed by complementation) as done in the proof of Proposition 7.

Again, elimination of nondeterminism can be performed thanks to Savitch Theorem [24]. So, non equivalence problem is in ExpSpace. Since coExpSpace= ExpSpace (simply because ExpSpace refers to a deterministic class of Turing machines), the equivalence problem is in ExpSpace. ◻

5 Concluding Remarks

We have introduced an automata-based method for computing the temporal equilibrium models of an arbitrary temporal theory, under the syntax of Linear-time Temporal Logic (LTL). This construction has allowed us solving several open problems about Temporal Equilibrium Logic (TEL) and its monotonic basis Temporal Here-and-There (THT). In particular, we were able to prove that THT satisfiability can be solved in PSPACE and is PSPACE-hard whereas TEL satisfiability is decidable (something not proven before) being solvable in EXPSPACE and at least PSPACE-hard (filling the gap is part of future work). Our method consists in constructing a Büchi automaton that captures all the temporal equilibrium models of an arbitrary theory. This also implies that the set of TEL models of any theory is ω-regular.

A recent approach [2,6] has developed a tool, called `STeLP`, that also captures TEL models of a theory in terms of a Büchi automaton. Our current proposal, however, has some important advantages. First, `STeLP` restricts the input syntax to so-called *splitable temporal logic programs*, a strict subclass of a normal form for TEL that further requires the introduction of auxiliary atoms for removing U and R operators, using a structure preserving transformation. On the contrary, our current method has no syntactic restrictions and directly works on the alphabet of the original theory, for which no transformation is required prior to the automaton construction. Second, once the `STeLP` input is written in the accepted syntax, it translates the input program into LTL by the addition of a set of formulae (the so-called *loop formulae*) whose number is, in the worst case, exponential on the size of the input. Future work includes the implementation of our current method as well a comparison in terms of efficiency with respect to the tool `STeLP`.

References

1. F. Aguado, P. Cabalar, G. Pérez, and C. Vidal. Strongly equivalent temporal logic programs. In *JELIA'08*, volume 5293 of *LNCS*, pages 8–20, 2008.
2. F. Aguado, P. Cabalar, G. Pérez, and C. Vidal. Loop formulas for splitable temporal logic programs. In *LPNMR'11*, volume 6645 of *LNCS*, pages 80–92, 2011.
3. G. Boenn, M. Brain, M. D. Vos, and J. Fitch. ANTON: Composing logic and logic composing. In *LPNMR'09*, volume 5753 of *LNCS*, pages 542–547, 2009.
4. R. Büchi. On a decision method in restricted second-order arithmetic. In *Intl. Congress on Logic, Method and Philosophical Science'60*, pages 1–11, 1962.
5. P. Cabalar. A normal form for linear temporal equilibrium logic. In *JELIA'10*, volume 6341 of *LNCS*, pages 64–76. Springer, 2010.
6. P. Cabalar and M. Diéguez. STELP - a tool for temporal answer set programming. In *LPNMR'11*, volume 6645 of *LNCS*, pages 370–375, 2011.
7. S. Demri and P. Gastin. Specification and verification using temporal logics. In *Modern applications of automata theory*, volume 2 of *IIsc Research Monographs*. World Scientific, 2011. to appear.
8. P. Ferraris. Answer sets for propositional theories. In *LPNMR'05*, volume 3662 of *LNCS*, pages 119–131, 2005.

9. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *ICLP'88*, pages 1070–1080. MIT Press, Cambridge, MA, 1988.
10. G. Grasso, S. Iiritano, N. Leone, V. Lio, F. Ricca, and F. Scalise. An ASP-based system for team-building in the Gioia-Tauro seaport. In *Proc. of the 12th Intl. Symposium on Practical Aspects of Declarative Languages (PADL 2010)*, volume 5937 of *LNCS*, pages 40–42. Springer, 2010.
11. A. Heyting. Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse*, pages 42–56, 1930.
12. H. Kautz. The logic of persistence. In *AAAI'86*, pages 401–405, 1986.
13. N. Leone, T. Eiter, W. Faber, M. Fink, G. Gottlob, and G. Greco. Boosting information integration: The INFOMIX system. In *Proc. of the 13th Italian Symposium on Advanced Database Systems, SEBD 2005*, pages 55–66, 2005.
14. Z. Manna and A. Pnueli. A hierarchy of temporal properties. In *PODC'90*, pages 377–408. ACM Press, 1990.
15. V. Marek and M. Truszczyński. *Stable models and an alternative logic programming paradigm*, pages 169–181. Springer-Verlag, 1999.
16. J. McCarthy. Elaboration tolerance. In *Proc. of the 4th Symposium on Logical Formalizations of Commonsense Reasoning (Common Sense 98)*, pages 198–217, London, UK, 1998.
17. J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence Journal*, 4:463–512, 1969.
18. I. Niemelä. Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence*, 25:241–273, 1999.
19. M. Nogueira, M. Balduccini, M. Gelfond, R. Watson, and M. Barry. An A-Prolog decision support system for the space shuttle. In *Proc. of the 1st Intl. Workshop on Answer Set Programming (ASP'01)*, 2001.
20. D. Pearce. A new logical characterisation of stable models and answer sets. In *NMELP'96*, volume 1216 of *LNAI*, pages 57–70. Springer, 1996.
21. D. Perrin and J.-E. Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Elsevier, 2004.
22. A. Pnueli. The temporal logic of programs. In *FOCS'77*, pages 46–57. IEEE, 1977.
23. S. Safra. *Complexity of Automata on Infinite Objects*. PhD thesis, The Weizmann Institute of Science, Rehovot, 1989.
24. W. Savitch. Relationships between nondeterministic and deterministic tape complexities. *JCSS*, 4(2):177–192, 1970.
25. A. Sistla and E. Clarke. The complexity of propositional linear temporal logic. *JACM*, 32(3):733–749, 1985.
26. A. Sistla, M. Vardi, and P. Wolper. The complementation problem for Büchi automata with applications to temporal logic. *TCS*, 49:217–237, 1987.
27. M. Vardi. Alternating automata: unifying truth and validity checking for temporal logics. In *CADE'97*, volume 1249 of *LNCS*, pages 191–206. Springer, 1997.
28. M. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *JCSS*, 32:183–221, 1986.
29. M. Vardi and P. Wolper. Reasoning about infinite computations. *I & C*, 115:1–37, 1994.
30. I. Černá and R. Pelánek. Relating hierarchy of linear temporal properties to model checking. In *MFCS'03*, volume 2747, pages 318–327. Springer, 2003.