Strong Equivalence of Non-monotonic Temporal Theories (extended version)*

Pedro Cabalar and Martín Diéguez

Dept. of Computer Science
University of Corunna, SPAIN
{cabalar,martin.dieguez}@udc.es

Abstract

Temporal Equilibrium Logic (TEL) is a non-monotonic temporal logic that extends Answer Set Programming (ASP) by introducing modal operators as those considered in Linear-time Temporal Logic (LTL). TEL allows proving temporal properties of ASP-like scenarios under the hypothesis of infinite time while keeping decidability. Formally, it extends *Equilibrium Logic* (the best-known logical formalisation of ASP) and, as the latter, it selects some models of a monotonic basis: the logic of *Temporal Here-and-There* (THT). In this paper we solve a problem that remained unanswered for the last six years: we prove that equivalence in the logic of THT is not only a sufficient, but also a *necessary* condition for strong equivalence of two TEL theories. This result has both theoretical and practical consequences. First, it reinforces the need of THT as a suitable monotonic basis for TEL. Second, it has allowed constructing a tool, ABSTEM, that can be used to check different types of equivalence between two arbitrary temporal theories. More importantly, when the theories are not THT-equivalent, the system provides a context theory that makes them behave differently, together with a Büchi automaton showing the temporal stable models that arise from that difference.

Introduction

Originally motivated by representational problems in action theories, the relation between Temporal and Non-Monotonic Reasoning (NMR) has historically played a central role in Knowledge Representation. Classical action languages such as *Situation Calculus* (McCarthy and Hayes 1969) or *Event Calculus* (Kowalski and Sergot 1986) have combined some NMR technique, usually predicate circumscription (McCarthy 1980), with a first-order formalisation of time using temporal predicates and objects (situations or events, respectively). Another way of dealing with temporal reasoning in NMR approaches has been using modal temporal logic, a combination perhaps less popular, but not unfrequent in the literature (Castilho, Gasquet, and Herzig 1999; Giordano, Martelli, and Schwind 2000; Baral and Zhao 2007). But probably, the simplest treatment of time we find in action theories is the use of an integer index to denote situations, as done for instance in the family of action languages (Gelfond and Lifschitz 1998). Focused on transition systems, these languages came from a method proposed in (Gelfond and Lifschitz 1993) for reasoning about actions using logic programs under the stable model semantics (Gelfond and Lifschitz 1988).

With the consolidation of Answer Set Programming (ASP) (Niemelä 1999; Marek and Truszczyński 1999) as a successful paradigm for practical knowledge representation, many examples and benchmarks formalising dynamic scenarios became available. ASP inherited the treatment of time from (Gelfond and Lifschitz 1998) action languages but further restricted all reasoning tasks to finite narratives, something required for grounding time-related variables. As a piece of example, consider an extremely simple ASP program where a fluent p represents that a switch is on and q represents that it is off. Moreover, suppose we have freedom to add p arbitrarily at any moment and that that either p or q holds initially. A typical ASP representation of this problem could look like this:

$$p(0) \vee q(0) \tag{1}$$
$$p(I{+}1) \leftarrow p(I), not\, q(I{+}1), sit(I) \tag{2}$$
$$q(I{+}1) \leftarrow q(I), not\, p(I{+}1), sit(I) \tag{3}$$
$$p(I) \vee not\, p(I) \leftarrow sit(I) \tag{4}$$

where (1) describes the initial state, (2) and (3) are the inertia rules for p and q, and (4) acts as a choice rule[1] allowing the introduction of p at any situation. Predicate sit would have some finite domain $0 \dots n$ for some constant $n \geq 0$. This approach is suitable, for instance, for solving planning problems incrementally (Gebser, Sabuncu, and Schaub 2011), using an iterative deepening strategy similar to SAT-based planning (Kautz and Selman 1992). In this way, to generate a plan for reaching a state in which we reach $p \wedge \neg q$, we would include two constraints for the last situation:

$$\bot \leftarrow not\, p(n) \tag{5}$$
$$\bot \leftarrow q(n) \tag{6}$$

and go increasing n until a solution is found. However, this strategy falls short for many temporal reasoning problems

*This research was partially supported by Spanish MEC project TIN2009-14562-C05-04.

[1] Generally speaking, a disjunction of the form $\varphi \vee not\, \varphi$ in ASP is not a tautology. When included in a rule head it is usually written as $\{\ \varphi\ \}$ and acts as a non-deterministic choice possibly allowing the derivation of φ.

that involve dealing with infinite time such as proving the non-existence of a plan or checking the satisfaction of temporal properties of a given dynamic system. For instance, questions such as "is there a reachable state in which both p and q are false?" or "can we show that whenever p is true it will remain so forever?" can be answered by an analytical inspection of our simple program, but cannot be solved in an automated way.

To overcome these limitations, (Cabalar and Pérez 2007) proposed a temporal extension of *Equilibrium Logic* (Pearce 1996), the best-known logical formalisation of ASP. This extension, which received the name of *Temporal Equilibrium Logic* (TEL), is defined as follows. First, it extends the monotonic basis of Equilibrium Logic, the intermediate logic of *Here-and-There* (HT) (Heyting 1930), by introducing the full syntax of the well-known *Linear-time Temporal Logic* (LTL) (Kamp 1968; Pnueli 1977). The result of this combination is called *Temporal Here-and-There* (THT). Then, a selection criterion on THT models is imposed, obtaining non-monotonicity in this way. As a result, TEL constitutes a full non-monotonic temporal logic that allows a proper definition of *temporal stable models* for any arbitrary theory in the syntax of LTL.

Following our example, the ASP program (1)-(4) would be represented in TEL as:

$$p \vee q \tag{7}$$
$$\Box(p \wedge \neg \bigcirc q \rightarrow \bigcirc p) \tag{8}$$
$$\Box(q \wedge \neg \bigcirc p \rightarrow \bigcirc q) \tag{9}$$
$$\Box(p \vee \neg p) \tag{10}$$

where, as usual in LTL, '$\Box$' stands for "always" and '$\bigcirc$' stands for "next." Checking whether p and q can be eventually false would correspond to look for a plan satisfying the constraint:

$$\neg\Diamond(\neg p \wedge \neg q) \rightarrow \bot \tag{11}$$

with '$\Diamond$' meaning "eventually." Similarly, to test whether p remains true after becoming true we would add the constraint:

$$\Box(p \rightarrow \Box p) \rightarrow \bot \tag{12}$$

and check that, indeed, no temporal stable model exists.

In the past years, several interesting results about TEL were obtained – see survey (Aguado et al. 2013). For instance, a tool called `STeLP` (Cabalar and Diéguez 2011) allowed computing the temporal stable models for a syntactic subset of TEL, the so-called *splittable* temporal logic programs, covering most dynamic scenarios in the ASP literature. A temporal program is *splittable* if it consists of ASP rules possibly prefixed by $\Box$ and whose literals may include a $\bigcirc$ operator but never introduce a dependence backwards in time (the rule consequent is never placed on the past of any literal in the antecedent). In other words, a splittable program would correspond to a *transition system* with a Markov property (the resulting state only depends on the previous state). As an example, rules (7)-(10) are all splittable. `STeLP` further allowed arbitrary temporal constraints of the form $\varphi \rightarrow \bot$ such as, for instance, (11), (12). Non-splittable formulas not covered by the syntax of `STeLP` could be, for instance:

$$\neg p \rightarrow \Diamond q \tag{13}$$
$$\Box\Diamond r \rightarrow s \tag{14}$$

where (13) would mean that if p cannot be proved now then q becomes eventually true, whereas (14) points out that if r is true infinitely often then s is true now.

Arbitrary theories have been studied in (Cabalar and Demri 2011), where authors provide an algorithm (that had not been implemented up to date), based on several transformations on Büchi automata (a variation of finite automata that accept languages with infinite length words) to obtain a final one which captures the temporal stable models of an input theory.

Despite of all these advances, several important questions about TEL remained unsolved. One of them has to do with the property of *strong equivalence* in TEL. In NMR, the regular equivalence, understood as a mere coincidence of selected models, is too weak to consider that one theory Γ_1 can be safely replaced by a second one Γ_2 since the addition of a context Γ may make them behave in a different way due to non-monotonicity. Formally, we say that Γ_1 and Γ_2 are *strongly equivalent* when, for any arbitrary theory Γ, both $\Gamma_1 \cup \Gamma$ and $\Gamma_2 \cup \Gamma$ have the same selected models (in this case, stable models). (Lifschitz, Pearce, and Valverde 2001) proved that checking equivalence in the logic of Here-and-There is a necessary and sufficient condition for strong equivalence in Equilibrium Logic, that is, Γ_1 and Γ_2 are strongly equivalent iff $\Gamma_1 \equiv_{HT} \Gamma_2$. A pair of strong equivalence checkers are, for instance, (Valverde 2004) and (Chen, Lin, and Li 2005) respectively based on HT-tableaux and the translation into SAT proposed in (Lin 2002). This result for propositional HT was further extended to arbitrary first-order theories in (Lifschitz, Pearce, and Valverde 2007). It must be noticed that one direction of this result, the sufficient condition, is actually trivial. As HT is monotonic, $\Gamma_1 \equiv_{HT} \Gamma_2$ implies $\Gamma_1 \cup \Gamma \equiv_{HT} \Gamma_2 \cup \Gamma$ and so, their selected models will also coincide. The real significant result is the opposite direction, namely, that HT-equivalence is also a necessary condition for strong equivalence, as it shows that HT is strong enough as a monotonic basis for Equilibrium Logic.

In the case of TEL, (Aguado et al. 2008) implemented a prototype checker and used it on some examples exploiting the trivial direction, i.e., that THT-equivalence is obviously a sufficient condition for strong equivalence in TEL. However, during the past six years, the question whether THT-equivalence was also necessary or not remained unanswered. This raised doubts on the adequacy of THT as a basis for TEL and had also some practical negative consequences. In particular, when two theories Γ_1, Γ_2 were not THT-equivalent, the checker could not answer anything about strong equivalence, while one would expect to be provided with a negative answer plus some context Γ that made them behave differently.

In this paper we adapt a result from (Lifschitz, Pearce, and Valverde 2007) to prove that indeed THT-equivalence is a necessary condition for TEL strong equivalence and use this proof, combined with previous theoretical results, to construct a tool, ABSTEM[2], that allows the formal study of arbitrary temporal theories in different ways. First, it implements the technique in (Cabalar and Demri 2011) to compute the temporal stable models of an arbitrary theory Γ, displaying them as a Büchi automaton. Second, given two theories Γ_1 and Γ_2, it allows checking different types of equivalence: LTL-equivalence, weak equivalence (coincidence in temporal stable models) and THT-equivalence which, as said before, corresponds to strong equivalence. When a negative answer for strong equivalence is obtained, the tool also suggests a context formula Γ that makes Γ_1 and Γ_2 behave differently and shows either one or all temporal stable models in the difference, again in the form of Büchi automata.

The rest of the paper is organised as follows. In the next section we recall the basic definitions of Temporal Equilibrium Logic. After that, we overview the automata-based techniques from (Cabalar and Demri 2011). The next section contains the main theorem showing that THT-equivalence is a necessary condition for strong equivalence in TEL. This is followed by an explanation of the implementation together with a practical example. Finally, we include some conclusions and future work.

Temporal Equilibrium Logic

We begin defining the (monotonic) logic of THT as follows. The syntax is defined as in propositional LTL. A temporal *formula* φ can be expressed following the grammar shown below:

$$\varphi ::= \bot \mid p \mid \alpha \wedge \beta \mid \alpha \vee \beta \mid \alpha \rightarrow \beta \mid \bigcirc\alpha \mid \alpha\, \mathcal{U}\, \beta \mid \alpha\, \mathcal{R}\, \beta$$

where p is an atom of some finite signature At, and α and β are temporal formulas in their turn.

The formula $\alpha\, \mathcal{U}\, \beta$ stands for "α *until* β" whereas $\alpha\, \mathcal{R}\, \beta$ is read as "α *release* β" and is the dual of "until." Derived operators such as $\Box$ ("always") and $\Diamond$ ("at some future time") are defined as $\Box\varphi \stackrel{\text{def}}{=} \bot\, \mathcal{R}\, \varphi$ and $\Diamond\varphi \stackrel{\text{def}}{=} \top\, \mathcal{U}\, \varphi$. Other usual propositional operators are defined as follows: $\neg\varphi \stackrel{\text{def}}{=} \varphi \rightarrow \bot$, $\top \stackrel{\text{def}}{=} \neg\bot$ and $\varphi \leftrightarrow \psi \stackrel{\text{def}}{=} (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.

Given a finite propositional signature At, an LTL-*interpretation* $\mathbf{T}$ is an infinite sequence of sets of atoms, $T_0, T_1, \ldots$ with $T_i \subseteq At$ for all $i \geq 0$. Given two LTL-interpretations $\mathbf{H}, \mathbf{T}$ we define $\mathbf{H} \leq \mathbf{T}$ as: $H_i \subseteq T_i$ for all $i \geq 0$. A THT-*interpretation* $\mathbf{M}$ for At is a pair of LTL-interpretations $\langle \mathbf{H}, \mathbf{T} \rangle$ satisfying $\mathbf{H} \leq \mathbf{T}$. A THT-interpretation is said to be *total* when $\mathbf{H} = \mathbf{T}$.

Definition 1 (Satisfaction) *We define when an interpretation* $\mathbf{M} = \langle \mathbf{H}, \mathbf{T} \rangle$ satisfies *a formula* φ *at a state* $i \geq 0$, *written* $\mathbf{M}, i \models \varphi$, *recursively as follows:*

1. $\mathbf{M}, i \models p$ *iff* $p \in H_i$, *with p an atom.*
2. $\wedge, \vee, \bot$ *as usual*
3. $\mathbf{M}, i \models \varphi \rightarrow \psi$ *iff for all* $x \in \{\mathbf{H}, \mathbf{T}\}$, $\langle x, \mathbf{T} \rangle, i \not\models \varphi$ *or* $\langle x, \mathbf{T} \rangle, i \models \psi$.
4. $\mathbf{M}, i \models \bigcirc\, \varphi$ *iff* $\mathbf{M}, i{+}1 \models \varphi$
5. $\mathbf{M}, i \models \varphi\, \mathcal{U}\, \psi$ *iff* $\exists k \geq i$ *such that* $\mathbf{M}, k \models \psi$ *and* $\forall j \in \{i, \ldots, k\text{-}1\}, \mathbf{M}, j \models \varphi$.
6. $\mathbf{M}, i \models \varphi\, \mathcal{R}\, \psi$ *iff* $\forall k \geq i$ *such that* $\mathbf{M}, k \not\models \psi$ *then* $\exists j \in \{i, \ldots, k\text{-}1\}, \mathbf{M}, j \models \varphi$. ■

We say that $\langle \mathbf{H}, \mathbf{T} \rangle$ is a *model* of a theory Γ, written $\langle \mathbf{H}, \mathbf{T} \rangle \models \Gamma$, iff $\langle \mathbf{H}, \mathbf{T} \rangle, 0 \models \alpha$ for all formulas $\alpha \in \Gamma$. It is easy to see that restricting the study to total interpretations, THT-satisfaction collapses to LTL-satisfaction, i.e.:

Proposition 1 (from (Aguado et al. 2013)) $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \varphi$ *in THT iff* $\mathbf{T}, i \models \varphi$ *in LTL.*

The following property, called *persistence*, can be easily proved by structural induction.

Proposition 2 (from (Aguado et al. 2013)) $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \varphi$ *implies* $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \varphi$.

An interpretation $\mathbf{M}$ is a *temporal equilibrium model* of a theory Γ if it is a total model of Γ, that is, $\mathbf{M} = \langle \mathbf{T}, \mathbf{T} \rangle \models \Gamma$, and there is no $\mathbf{H} < \mathbf{T}$ such that $\langle \mathbf{H}, \mathbf{T} \rangle \models \Gamma$. An LTL-interpretation $\mathbf{T}$ is a *temporal stable model* (TS-model) of a theory Γ iff $\langle \mathbf{T}, \mathbf{T} \rangle$ is a temporal equilibrium model of Γ. By Proposition 1 it is easy to see that:

Observation 1 *Any TS-model of a temporal theory* Γ *is also an LTL-model of* Γ. ■

As happens in LTL, the set of TS-models of a theory Γ can be captured by a Büchi automaton (Büchi 1962), a kind of ω-automaton (that is, a finite automaton that accepts words of infinite length). In this case, the alphabet of the automaton would be the set of states (classical propositional interpretations) and the acceptance condition is that a word (a sequence of states) is accepted iff it corresponds to a run of the automaton that visits some acceptance state an infinite number of times. As an example, Figure 1 shows the TS-models for the theory (7)-(10) which coincide with sequences of states of the forms $\{q\}^*\{p\}^\omega$ or $\{q\}^\omega$. Notice how p and q are never true simultaneously, whereas once p becomes true, it remains true forever.

It is important to note that, as happens for regular finite automata, Büchi automata are also closed under complementation, union, intersection and renaming operations, all of them involved in the process of computing the temporal equilibrium models of a temporal formula we recall next.

Computing Temporal Stable Models of Arbitrary Theories

As explained before, the tool ABSTEM constitutes the first implementation capable of computing TS-models for arbitrary temporal formulas, without syntactic restrictions. This implementation is based on the method proposed in (Cabalar and Demri 2011) used there as a formal tool to prove decidability of THT and TEL plus some complexity results. The

[2] http://kr.irlab.org/?q=abstem

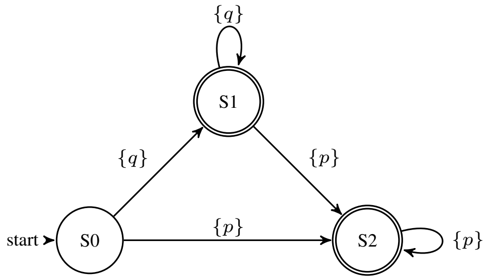


Figure 1: Temporal stable models of theory (7)-(10).

method obtains the TS-models of a formula φ by performing several operations on a pair of automata derived from φ. The first automaton, denoted as A_φ, accepts the total THT-models $\langle \mathbf{T}, \mathbf{T} \rangle$ of φ. By Proposition 1 this simply amounts to compute the LTL models $\mathbf{T}$ of φ using an automata construction method for LTL like (Vardi and Wolper 1986). The second automaton, denoted as $A_{\varphi''}$, accepts the non-total THT-models $\langle \mathbf{H}, \mathbf{T} \rangle$ of φ. The final set of TS-models is obtained from the composition $A_\varphi \cap \overline{h(A_{\varphi''})}$ where $h(A_{\varphi''})$ filters out the $\mathbf{H}$ component of non-total models, $\overline{h(A_{\varphi''})}$ is the negation of $h(A_{\varphi''})$ and finally $\cap$ denotes the automata product.

The computation of $A_{\varphi''}$ is done exploiting a translation of THT into LTL first presented in (Aguado et al. 2008) and directly extrapolating the translation of HT into classical logic in (Pearce, Tompits, and Woltran 2001). This translation relies on extending the signature At with a new set of atoms $\{p' \mid p \in At\}$ so that p' represents the truth of p in $\mathbf{H}$ while its non-primed version p is used for $\mathbf{T}$. The translation of φ, written φ^*, is recursively defined as follows:

- $(\bot)^* \stackrel{\text{def}}{=} \bot$
- $(p)^* \stackrel{\text{def}}{=} p'$, for any atom $p \in At$
- $(\alpha \to \beta)^* \stackrel{\text{def}}{=} (\alpha \to \beta) \wedge (\alpha^* \to \beta^*)$
- $(\bigcirc\alpha)^* \stackrel{\text{def}}{=} \bigcirc\alpha^*$
- $(\alpha \odot \beta)^* \stackrel{\text{def}}{=} \alpha^* \odot \beta^*$, with $\odot \in \{\wedge, \vee, \mathcal{U}, \mathcal{R}\}$

To impose the restriction $\mathbf{H} \leq \mathbf{T}$ we further include the axioms:

$$\bigwedge_{p \in At} \Box(p' \to p) \qquad \textbf{(Ax1)}$$

(Aguado et al. 2008) proved that LTL-models of formula $\varphi^* \wedge$(**Ax1**) are in one-to-one correspondence to THT-models of φ. To obtain non-total THT-models, that is $\mathbf{H} < \mathbf{T}$, we further strengthen the formula adding this axiom:

$$\bigvee_{p \in At} \Diamond(\neg p' \wedge p) \qquad \textbf{(Ax2)}$$

intuitively meaning that there is some $p \in At$ for which, eventually, p is true in $\mathbf{T}$ but not in $\mathbf{H}$. Automaton $A_{\varphi''}$ is built from the LTL formula $\varphi'' \stackrel{\text{def}}{=} \varphi^* \wedge$ (**Ax1**) $\wedge$ (**Ax2**).

As the input alphabets of A_φ and $A_{\varphi''}$ are different, the intersection of their languages would not be possible without filtering the latter. Filtering $A_{\varphi''}$ consists in removing the atoms p' from its transitions and obtaining a new automaton, $h(A_{\varphi''})$. This automaton captures the $\mathbf{T}$-components of non-total models; in this way, its complementary automaton $\overline{h(A_{\varphi''})}$ accepts the $\mathbf{T}$ sequences that do not form a non-total model, but perhaps they are not models either. Thus, the final product $A_\varphi \cap \overline{h(A''_\varphi)}$ captures those $\mathbf{T}$ such that $\langle \mathbf{T}, \mathbf{T} \rangle$ is a total model of φ and no non-total model $\langle \mathbf{H}, \mathbf{T} \rangle$ can be formed.

As an example, let us take again the (non-splittable) formula $\varphi =$(13). The translation φ^* corresponds[3] to:

$$(\neg p \to \Diamond q) \wedge (\neg p \wedge \neg p' \to \Diamond q') \qquad (15)$$

and the intermediate automata A_φ, $A_{\varphi''}$, $h(A_{\varphi''})$, $\overline{h(A''_\varphi)}$ and $A_\varphi \cap \overline{h(A''_\varphi)}$ are shown in Figures 2(a), 2(b), 2(c), 2(d) and 2(e), respectively. The latter shows, indeed, the TS-models of φ, where p is left false forever by default (there is no evidence for p) and q becomes true at some punctual future state, but only once due to truth minimality.

`ABSTEM` uses a modified version of the `SPoT`[4] library as a backend. This library provides an extensive API to deal with several types of ω-automata allowing, for instance, their complementation, intersection or even minimisation, for allowing a more compact representation. It also provides several methods for obtaining ω-automata from an LTL formula.

Temporal Strong Equivalence

The second important feature of `ABSTEM` is the possibility of checking strong equivalence for two arbitrary temporal theories[5] or formulas Γ_1 and Γ_2. This introduces, in fact, a temporal dimension in the study of strong equivalence for ASP, since it means showing that Γ_1 and Γ_2 have the same behaviour not only under any hypothetical context, but also for narratives with unlimited time. Of course, resorting to temporal logic is not always necessary. For instance, when we just deal with transition systems, a straightforward possibility[6] is to restrict the study to transitions between two consecutive states, say 0 and 1, using the non-temporal approach to check strong equivalence (that is, HT-equivalence). However, when non-splittable formulas are involved, such as (11)-(14), that technique is not possible any more. As an example of the kind of difficulties we find when we move to non-splittable theories, consider the theory Γ_2:

$$\Box(p \wedge q \quad \to \quad \bigcirc q) \qquad (16)$$
$$\Box(\neg p \wedge \bigcirc p \quad \to \quad \bigcirc\bigcirc q) \qquad (17)$$
$$p \quad \to \quad \bot \qquad (18)$$
$$\Box(p \wedge q \quad \to \quad \bot) \qquad (19)$$

[3]Remember that $\neg p = (p \to \bot)$.

[4]`http://spot.lip6.fr/wiki/`

[5]For simplicity, we assume finite theories and we indistinctly represent them as the conjunction of their formulas.

[6]Suggested by Joohyung Lee during conference LPNMR'11.

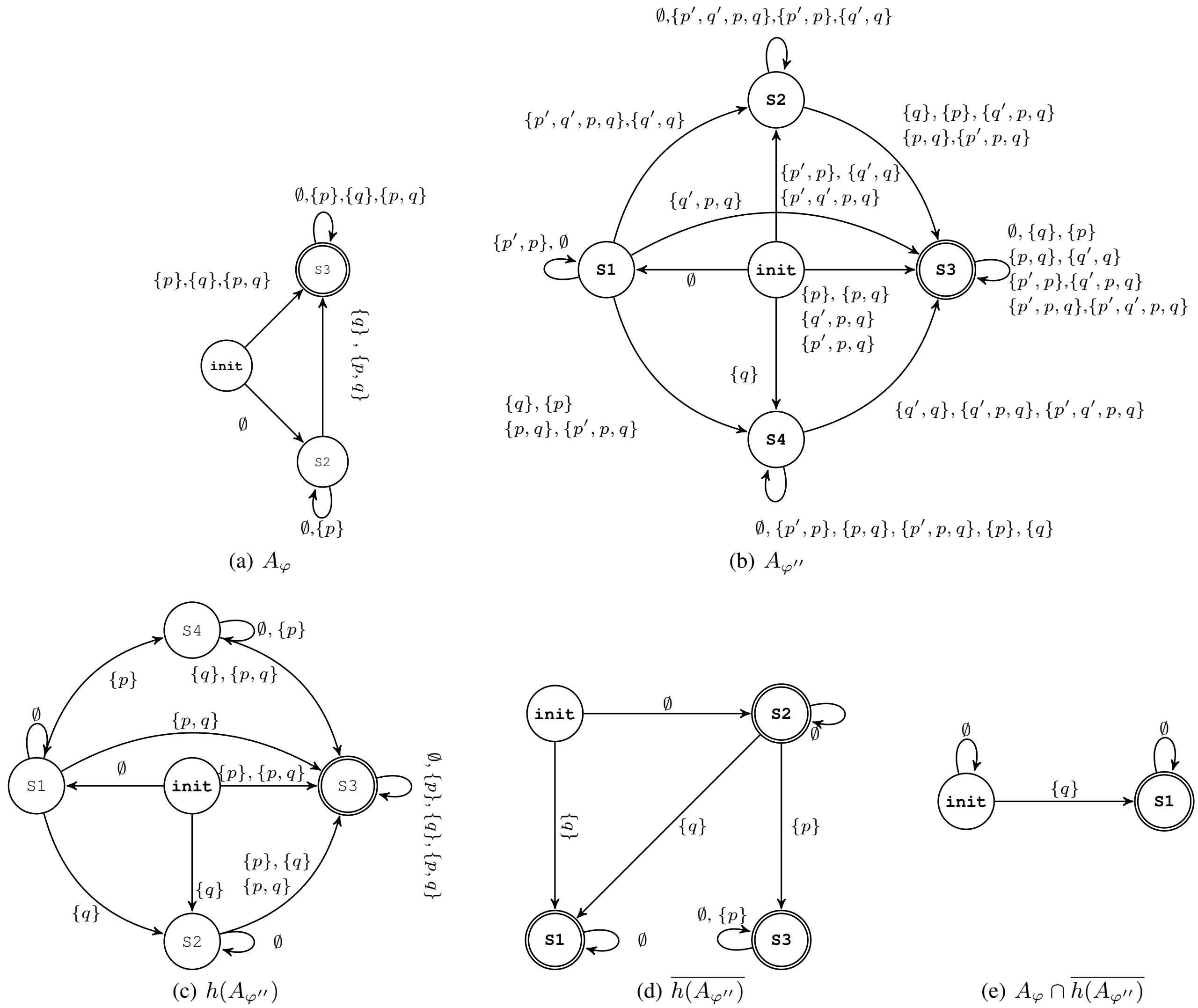


Figure 2: Intermediate automata generated in the example $\varphi = \neg p \rightarrow \Diamond q$.

where (17) is non-splittable, since it checks atoms at three different situations. As the rule is preceded by $\Box$, if we wanted to make a static analysis, we should take transitions for states $0, 1, 2$ but also for $1, 2, 3$. This should be carefully chosen by hand and, in any case, it is difficult to conclude, for instance, that in the theory above, the first two formulas (16) and (17) can be safely replaced by:

$$\Box(p \rightarrow \bigcirc q) \tag{20}$$

as stated below.

Proposition 3 *Theory* $\Gamma_3 =$ (18)-(20) *is strongly equivalent to* $\Gamma_2 =$(16)-(19). ■

In fact, if we check these two theories using ABSTEM it just provides a positive answer and no further explanation is required. The proof for Proposition 3 (see the Appendix) relies on the fact that Γ_2 and Γ_3 are THT-equivalent and, as we explained before, this is trivially a sufficient condition for strong equivalence (see (Aguado et al. 2008)).

Theorem 1 (Sufficient condition) *If two temporal formulas* α *and* β *are THT-equivalent then they are strongly equivalent in TEL.* ■

Until now, however, we missed the other direction, namely, that THT-equivalence is also a necessary condition for strong equivalence. In the rest of this section, we prove this result adapting the main proof in (Lifschitz, Pearce, and Valverde 2007). Before introducing the main theorem we will first introduce some auxiliary results that will be used for the main proof and for implementation purposes. We begin defining the following axiom:

$$\gamma_0 \stackrel{\text{def}}{=} \bigwedge_{p \in At} \Box(p \vee \neg p)$$

The effect of adding this axiom to a theory is restricting to total models, as stated below:

Proposition 4 *Let $\langle \mathbf{H}, \mathbf{T} \rangle$ be a THT interpretation for signature At. If $\langle \mathbf{H}, \mathbf{T} \rangle \models \gamma_0$ then $\mathbf{H} = \mathbf{T}$.*

Corollary 1 *For any formula α for signature At, the LTL-models of $\alpha \wedge \gamma_0$ coincide with its TS-models.* ■

Lemma 1 *Let α and β be two LTL-equivalent formulas and let $\gamma = (\beta \to \gamma_0)$. Then, the following conditions are equivalent:*

(i) *There exists some $\mathbf{H} < \mathbf{T}$ such that $\langle \mathbf{H}, \mathbf{T} \rangle \not\models \alpha \to \beta$;*

(ii) *$\mathbf{T}$ is TS-model of $\beta \wedge \gamma$ but not TS-model of $\alpha \wedge \gamma$.* ■

Theorem 2 (Main theorem: necessary condition) *If two temporal formulas α and β are strongly equivalent in TEL then they are THT-equivalent.* ■

Proof of Theorem 2. The proof follows (Lifschitz, Pearce, and Valverde 2007) although, for implementation purposes, it is more convenient here to prove the contraposition of the result, that is: if α and β are not THT-equivalent then there is some context formula γ for which $\alpha \wedge \gamma$ and $\beta \wedge \gamma$ have different TS-models.

Assume first that α and β have different total models, i.e., different LTL-models. Then, the LTL-models of $\alpha \wedge \gamma_0$ and $\beta \wedge \gamma_0$ also differ (because γ_0 is an LTL tautology). But by Corollary 1, LTL-models of these theories are exactly their TS-models, which therefore, also differ.

Suppose now that α and β are LTL-equivalent but not THT-equivalent. Then, there is some THT-countermodel $\langle \mathbf{H}, \mathbf{T} \rangle$ of either $(\alpha \to \beta)$ or $(\beta \to \alpha)$, and given LTL-equivalence of α and β, the countermodel is non-total, $\mathbf{H} < \mathbf{T}$. Without loss of generality, assume $\langle \mathbf{H}, \mathbf{T} \rangle \not\models \alpha \to \beta$. By Lemma 1, taking the formula $\gamma = (\beta \to \gamma_0)$, we get that $\mathbf{T}$ is TS-model of $\beta \wedge \gamma$ but not TS-model of $\alpha \wedge \gamma$. ■

Corollary 2 *Let α and β be two LTL-equivalent formulas and let $\gamma = (\beta \to \gamma_0)$. Then, any TS-model of $\alpha \wedge \gamma$ is a TS-model of $\beta \wedge \gamma$.* ■

Implementation and a practical example

The procedure for checking strong equivalence in ABSTEM is shown in Algorithm 1. It takes two arbitrary propositional temporal formulas α and β and returns either **true**, if they are strongly equivalent, or a triple with a formula γ and two automata A_1, A_2 otherwise. The meaning of this information is that A_1 captures TS-models of $\alpha \wedge \gamma$ that are not TS-models of $\beta \wedge \gamma$ and, analogously, A_2 captures TS-models of $\beta \wedge \gamma$ that are not TS-models of $\alpha \wedge \gamma$. The procedure uses several auxiliary routines: ltl_to_Büchi(φ) uses a SPoT function to obtain a Büchi automaton from an LTL-formula φ; one_path(A) returns a single path from an automaton A; finally $h(A)$ is the result of filtering out primed atoms from automaton A (as we explained in Section 3). The first three **if**'s in the algorithm check LTL-equivalence. These steps are trivial, except that when option 'compute_one' is selected and the theories are not LTL-equivalent, only one path of the first non-empty automaton is returned. As stated in the proof of the main theorem, when LTL-equivalence fails, we take γ_0 as context formula.

If α and β are LTL-equivalent, the algorithm proceeds to find all non-total $\mathbf{H} < \mathbf{T}$ countermodels of $\alpha \to \beta$. These non-total countermodels are captured by an automaton A for the formula $\neg(\alpha \to \beta)^* \wedge (\mathbf{Ax1}) \wedge (\mathbf{Ax2})$ using the same translation $(\cdot)^*$ and axioms we saw in Section 3 for computing non-total THT-models. If the language of this automaton is not empty, by Lemma 1, the $\mathbf{T}$ components of these non-total countermodels are precisely the TS-models of $\beta \wedge \gamma$ that are not TS-models of $\alpha \wedge \gamma$, for $\gamma = (\beta \to \gamma_0)$. To obtain those $\mathbf{T}$ components we compute $A_2 := h(A)$ that, as explained in Section 3, filters out the auxiliary atoms representing truth in $\mathbf{H}$. The algorithm then returns $\langle \gamma, \emptyset, A_2 \rangle$ since, by Corollary 2, there are no TS-models of $\alpha \wedge \gamma$ that are not TS-models of $\beta \wedge \gamma$. If, on the contrary, the language of A is empty, we proceed in the analogous way for the other direction $\beta \to \alpha$. Finally, if in both cases we get an empty automaton, then this means that any non-total interpretation is a model of $\alpha \leftrightarrow \beta$ something that, together with the LTL-equivalence of α and β, means that the two formulas are THT-equivalent, that is, strongly equivalent.

Algorithm 1 StrongEquivalenceTest(α, β)

Require: Two propositional temporal formulas α, β.
If option 'compute_one' is set, it returns just one TS-model when a difference is found.
Ensure: If α and β are THT-equivalent, it returns **true**. Otherwise, it returns a triple $\langle \gamma, A_1, A_2 \rangle$ where γ is a formula and A_1, A_2 are two automata such that:
A_1 captures TS-models($\alpha \wedge \gamma$) $\setminus$ TS-models($\beta \wedge \gamma$)
A_2 captures TS-models($\beta \wedge \gamma$) $\setminus$ TS-models($\alpha \wedge \gamma$).

```
A1 := ltl_to_Büchi(α ∧ ¬β)
if compute_one and A1 ≠ ∅ then
    return ⟨γ0, one_path(A1), ∅⟩
end if
A2 := ltl_to_Büchi(β ∧ ¬α)
if compute_one and A2 ≠ ∅ then
    return ⟨γ0, ∅, one_path(A2)⟩
end if
if A1 ≠ ∅ or A2 ≠ ∅ then
    return ⟨γ0, A1, A2⟩
end if
A=ltl_to_Büchi(¬(α → β)* ∧ (Ax1) ∧ (Ax2))
if A ≠ ∅ then
    A2 := h(A)
    if compute_one then
        A2 := one_path(A2)
    end if
    return ⟨(β → γ0), ∅, A2⟩
end if
A=ltl_to_Büchi(¬(β → α)* ∧ (Ax1) ∧ (Ax2))
if A ≠ ∅ then
    A1 := h(A)
    if compute_one then
        A1 := one_path(A1)
    end if
    return ⟨(α → γ0), A1, ∅⟩
end if
return true
```

As an example of use, let us take our previous "switch" example (7)-(10) and suppose we add a rule:

$$\Box(\neg p \rightarrow q) \tag{21}$$

trying to capture the idea that, when no information on p is available, q becomes true. This new rule is actually a new default for q that interacts with inertia rules (8),(9) destroying somehow their effect. Let β_1 be this extended theory, (7)-(10) plus (21). We can use ABSTEM to check the TS-models of β_1 (stored in file `beta1.abs`) as follows:

```
abstem -t -m -f beta1.abs
```

and we obtain the automaton in Figure 3(a) which corresponds to arbitrary sequences formed with states $\{p\}$ and $\{q\}$ – note the difference with respect to Figure 1 where p remained true after becoming true. This set of TS-models actually coincides with what one would expect from a formula of the form $\Box(p \vee q)$ since, as happens in ASP, truth minimality converts the disjunction into an exclusive or. Let us call $\alpha_1 = \Box(p \vee q)$. We check whether α_1 and β_1 have the same TS-models:

```
abstem -w -m -f alpha1.abs -f beta1.abs
```

and we obtain a positive answer. Furthermore, by a quick inspection on β_1 we can foresee that it is actually LTL-equivalent to α_1. First, in LTL (21) is equivalent to α_1, and (10) is just a tautology. The other three rules, (7)-(9) can be rewritten in LTL as $(p \vee q) \wedge \Box((p \vee q) \rightarrow \bigcirc(p \vee q))$ and this is equivalent to $\Box(p \vee q)$ too (it corresponds to the induction schema for $p \vee q$). We can use ABSTEM to confirm LTL-equivalence as follows:

```
abstem -l -m -f alpha1.abs -f beta1.abs
```

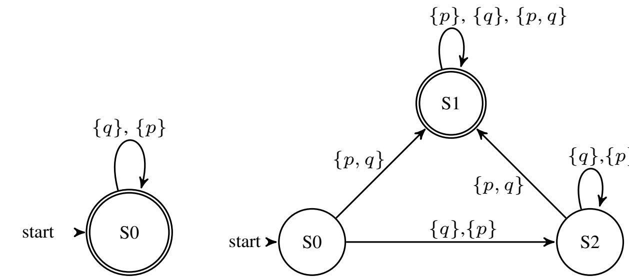
and we obtain again a positive answer. However, α_1 and β_1 are not THT-equivalent and so, they are not strongly equivalent. The strong equivalence checking in ABSTEM is done as follows:

```
abstem -s -m -f alpha1.abs -f beta1.abs
```

and the answer displayed this time is negative, containing this information:

```
Not strongly equivalent:
adding the context
( (p | q) & G(p | !p)
& G((p & !Xq)->Xp) & G((q & !Xp)->Xq)
& G(!p -> q))->
  (G(p | !p) & G(q | !q))
File seq_differences_0: (p & q){(q)}w
TS-model of beta1.abs but not alpha1.abs
```

The context formula just corresponds to $\gamma_1 = \beta_1 \rightarrow \gamma_0$ in ABSTEM syntax. The output already contains a path of states $\{p,q\}\{q\}^\omega$ that is a TS-model of $\beta_1 \wedge \gamma_1$ but not of $\alpha_1 \wedge \gamma_1$. The automaton in file `seq_differences_0` is shown in Figure 3(b) and captures all the TS-models of $\beta_1 \wedge \gamma_1$ that are not TS-models of $\alpha_1 \wedge \gamma_1$. Note that this contains all TS-models that differ since, by Corollary 2, TS-models of $\alpha_1 \wedge \gamma_1$ are also TS-models of $\beta_1 \wedge \gamma_1$.



(a) TS-models of α_1 and β_1.

(b) TS-models of $\beta_1 \wedge \gamma_1$ not of $\alpha_1 \wedge \gamma_1$.

Figure 3: Temporal stable models related to α_1 and β_1.

Conclusions

In this paper we have provided several results for equivalence checking in Temporal Equilibrium Logic (TEL), a non-monotonic modal logic that properly extends Answer Set Programming for arbitrary theories in the syntax of Linear-time Temporal Logic (LTL). As a first contribution, we have solved an open question related to the characterisation of strong equivalence, a type of equivalence from non-monotonic formalisms that asserts that two theories Γ_1, Γ_2 yield the same results even after the addition of any common context formula Γ. Adapting a result from (Lifschitz, Pearce, and Valverde 2007) to the temporal case, we were able to prove that equivalence in the monotonic basis of TEL, the logic of Temporal Here-and-There (THT), is not only a sufficient but also a *necessary* condition for strong equivalence in TEL. This is important from the theoretical perspective, since it consolidates the role of THT as a suitable monotonic basis for TEL. But it also has practical consequences, since the proof for this result provides a way to form a context theory Γ that shows how Γ_1 and Γ_2 can be forced to behave in a different way.

The second main contribution of the paper is the implementation of a system, ABSTEM, for analysing TEL arbitrary theories in different ways. ABSTEM constitutes the first implementation of the automata-based method proposed in (Cabalar and Demri 2011) to compute the temporal stable models (represented by a Büchi automaton) of arbitrary theories in the syntax of propositional LTL. This feature is further used when checking weak equivalence of two theories, so that ABSTEM can either answer that their temporal stable models coincide, or display temporal stable models for which they differ. Finally, using a translation from (Aguado et al. 2008), we have also implemented a THT-equivalence checking. Thanks to our main proof, when a negative answer is obtained, ABSTEM is able to suggest a context formula Γ that causes a different behaviour. Furthermore, when this difference is found, the techniques for computing temporal stable models are used this time to display one or all models in the difference. Regarding efficiency and scalability, this prototype works satisfactorily for small theories like the ones presented in the paper. Future work will incorporate a limited use of variables and grounding for arbitrary theories to make proofs for larger theories. Still, it must be noticed that THT-satisfaction is PSPACE-complete (Cabalar and Demri 2011).

The main open topic for future work is the study of different syntactic forms for the context formula generated by ABSTEM when strong equivalence fails. Right now, it is a formula where implications and temporal operators can be nested in an arbitrary way. This does not necessarily have a straightforward translation as an ASP rule using temporal variables. For instance, an interesting question would be which is the temporal analogous to the so-called *uniform equivalence* (Eiter et al. 2004), that is, strong equivalence when the context theory is a set of atoms.

References

[Aguado et al. 2008] Aguado, F.; Cabalar, P.; Pérez, G.; and Vidal, C. 2008. Strongly equivalent temporal logic programs. In *JELIA'08*, volume 5293 of *Lecture Notes in Computer Science*, 8–20.

[Aguado et al. 2013] Aguado, F.; Cabalar, P.; Diéguez, M.; Pérez, G.; and Vidal, C. 2013. Temporal equilibrium logic: a survey. *Journal of Applied Non-Classical Logics* 23(1-2):2–24.

[Baral and Zhao 2007] Baral, C., and Zhao, J. 2007. Non-monotonic temporal logics for goal specification. In *Proc. of the 20th Intl. Joint Conf. on Artificial Intelligence (IJCAI'07)*, 236–242.

[Büchi 1962] Büchi, R. 1962. On a decision method in restricted second-order arithmetic. In *Intl. Congress on Logic, Method and Philosophical Science'60*, 1–11.

[Cabalar and Demri 2011] Cabalar, P., and Demri, S. 2011. Automata-based computation of temporal equilibrium models. In Vidal, G., ed., *Proceedings of the 21st International Workshop on Logic Program Synthesis and Transformation (LOPSTR'11)*, Lecture Notes in Computer Science. Odense, Denmark: Springer.

[Cabalar and Diéguez 2011] Cabalar, P., and Diéguez, M. 2011. STELP - a tool for temporal answer set programming. In *LPNMR'11*, volume 6645 of *Lecture Notes in Computer Science*, 370–375.

[Cabalar and Pérez 2007] Cabalar, P., and Pérez, G. 2007. Temporal equilibrium logic: a first approach. In *Proc. of the 11th Intl. Conf. on Computer Aided Systems Theory, (EUROCAST'07)*, volume 4739 of *Lecture Notes in Computer Science*, 241–248.

[Castilho, Gasquet, and Herzig 1999] Castilho, M. A.; Gasquet, O.; and Herzig, A. 1999. Formalizing action and change in modal logic I: the frame problem. *Journal of Logic and Computation* 9(5):701–735.

[Chen, Lin, and Li 2005] Chen, Y.; Lin, F.; and Li, L. 2005. SELP - a system for studying strong equivalence between logic programs. In *LPNMR'05*, volume 3662 of *Lecture Notes in Computer Science*, 442–446.

[Eiter et al. 2004] Eiter, T.; Fink, M.; Tompits, H.; and Woltran, S. 2004. Simplifying logic programs under uniform and strong equivalence. In *Proc. of the 7th Intl. Conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR'04)*, volume 2923 of *Lecture Notes in Computer Science*. Springer. 87–99.

[Gebser, Sabuncu, and Schaub 2011] Gebser, M.; Sabuncu, O.; and Schaub, T. 2011. An incremental answer set programming based system for finite model computation. *AI Commun.* 24(2):195–212.

[Gelfond and Lifschitz 1988] Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In *ICLP'88*. Cambridge, MA: MIT Press. 1070–1080.

[Gelfond and Lifschitz 1993] Gelfond, M., and Lifschitz, V. 1993. Representing action and change by logic programs. *The Journal of Logic Programming* 17:301–321.

[Gelfond and Lifschitz 1998] Gelfond, M., and Lifschitz, V. 1998. Action languages. *Linköping Electronic Articles in Computer and Information Science* 3(16). (`http://www.ep.liu.se/ea/cis/1998/016`).

[Giordano, Martelli, and Schwind 2000] Giordano, L.; Martelli, A.; and Schwind, C. 2000. Ramification and causality in a modal action logic. *Journal of Logic and Computation* 10(5):625–662.

[Heyting 1930] Heyting, A. 1930. Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse* 42–56.

[Kamp 1968] Kamp, J. 1968. *Tense Logic and the Theory of Linear Order*. Ph.D. Dissertation, University of California at Los Angeles.

[Kautz and Selman 1992] Kautz, H. A., and Selman, B. 1992. Planning as satisfiability. In *Proc. of the European Conf. on Artificial Intelligence (ECAI'92)*, 359–363.

[Kowalski and Sergot 1986] Kowalski, R., and Sergot, M. 1986. A logic-based calculus of events. *New Generation Computing* 4:67–95.

[Lifschitz, Pearce, and Valverde 2001] Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly equivalent logic programs. *Computational Logic* 2(4):526–541.

[Lifschitz, Pearce, and Valverde 2007] Lifschitz, V.; Pearce, D.; and Valverde, A. 2007. A characterization of strong equivalence for logic programs with variables. In *LPNMR'07*, volume 4483 of *Lecture Notes in Computer Science*. Springer.

[Lin 2002] Lin, F. 2002. Reducing strong equivalence of logic programs to entailment in classical propositional logic. In Bradzil, P., and Jorge, A., eds., *Proc. of the 8th Intl. Conf. on Principles and Knowledge Representation and Reasoning (KR'02)*. Morgan Kaufmann. 170–176.

[Marek and Truszczyński 1999] Marek, V., and Truszczyński, M. 1999. *Stable models and an alternative logic programming paradigm*. Springer-Verlag. 169–181.

[McCarthy and Hayes 1969] McCarthy, J., and Hayes, P. 1969. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence Journal* 4:463–512.

[McCarthy 1980] McCarthy, J. 1980. Circumscription: A form of non-monotonic reasoning. *Artificial Intelligence* 13:27–39.

[Niemelä 1999] Niemelä, I. 1999. Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence* 25:241–273.

[Pearce, Tompits, and Woltran 2001] Pearce, D.; Tompits, H.; and Woltran, S. 2001. Encodings for equilibrium logic and logic programs with nested expressions. In *10th Portuguese Conf. on Artificial Intelligence, EPIA 2001*, volume 2258 of *Lecture Notes in Computer Science*, 306–320. Springer.

[Pearce 1996] Pearce, D. 1996. A new logical characterisation of stable models and answer sets. In *NMELP'96*, volume 1216 of *Lecture Notes in Artificial Intelligence*. Springer. 57–70.

[Pnueli 1977] Pnueli, A. 1977. The temporal logic of programs. In *FOCS'77*, 46–57. IEEE.

[Valverde 2004] Valverde, A. 2004. tabeql: A tableau based suite for equilibrium logic. In *JELIA'04*, volume 3229 of *Lecture Notes in Artificial Intelligence*, 734–737.

[Vardi and Wolper 1986] Vardi, M., and Wolper, P. 1986. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Sciences* 32:183–221.

Appendix. Proofs

Proof of Theorem 1. The proof is straightforward. If α and β are satisfied by the same THT-interpretations then $\alpha \wedge \gamma$ and $\beta \wedge \gamma$ (for every formula γ) are also satisfied by the same THT-interpretations; But then, selecting among them the temporal equilibrium models yields the same effect on both. ■

Proof of Proposition 3. Using Theorem 1, we will show that Γ_2 and Γ_3 are THT-equivalent. We begin recalling a pair of LTL theorems that are also valid in THT:

$$\Box(\alpha \wedge \beta) \leftrightarrow \Box\alpha \wedge \Box\beta \quad (22)$$

$$\Box\alpha \leftrightarrow \alpha \wedge \Box(\alpha \to \bigcirc\alpha) \quad (23)$$

$$\bigcirc(\alpha \to \beta) \leftrightarrow (\bigcirc\alpha \to \bigcirc\beta) \quad (24)$$

We will also use the HT-valid equivalences that allow unnesting implications:

$$\alpha \to (\beta \to \gamma) \leftrightarrow \alpha \wedge \beta \to \gamma \quad (25)$$

$$(\alpha \to \beta) \to \gamma \leftrightarrow (\neg\alpha \to \gamma) \wedge (\beta \to \gamma) \wedge(\alpha \vee \neg\beta \vee \gamma) \quad (26)$$

(23) is the induction schema: applied to (20) we get

$$\Box(p \to \bigcirc q) \leftrightarrow (p \to \bigcirc q) \wedge \Box((p \to \bigcirc q) \to \bigcirc(p \to \bigcirc q))$$

By (24) we get:

$$\leftrightarrow (p \to \bigcirc q) \wedge \Box((p \to \bigcirc q) \to (\bigcirc p \to \bigcirc\bigcirc q))$$

We unfold the nested implications using (25) and (26) and use (22) to distribute $\Box$ on conjunction afterwards, obtaining:

$$\leftrightarrow (p \to \bigcirc q) \wedge \Box(\neg p \wedge \bigcirc p \to \bigcirc\bigcirc q) \wedge\Box(\bigcirc q \wedge \bigcirc p \to \bigcirc\bigcirc q) \wedge\Box(\bigcirc p \to p \vee \neg\bigcirc q \vee \bigcirc\bigcirc q) \quad (27)$$

So we concluded that (20) is equivalent to (27). Now, from (19) it is easy to see that $\Box(p \to \neg q)$ and in particular, $\Box(\bigcirc p \to \neg\bigcirc q)$ something that implies the last conjunct of (27). On the other hand, as $\neg p$ follows from (18), we conclude that the following implications also hold:

$$p \to \bigcirc q \qquad q \wedge p \to \bigcirc q$$

so that we can also remove the first conjunct in (27) whereas the third one can be replaced by $\Box(q \wedge p \to \bigcirc q)$. To sum up, when constraints (18) and (19) are present, (27) is eventually equivalent to:

$$\Box(\neg p \wedge \bigcirc p \to \bigcirc\bigcirc q) \wedge \Box(q \wedge p \to \bigcirc q)$$

which is the conjunction of (16) and (17). ■

Proof of Proposition 4. We have to prove that $\forall i \in \mathbb{N}$, $H_i = T_i$. Since $H_i \subseteq T_i$ from $\mathbf{H} \leq \mathbf{T}$, we just need to prove $T_i \subseteq H_i$. Take some $p \in T_i$. Obviously, $\langle \mathbf{H}, \mathbf{T}\rangle, i \not\models \neg p$. But from $\langle \mathbf{H}, \mathbf{T}\rangle \models \Box(p \vee \neg p)$ we conclude $\langle \mathbf{H}, \mathbf{T}\rangle, i \models p \vee \neg p$ and so $\langle \mathbf{H}, \mathbf{T}\rangle, i \models p$ that is $p \in H_i$. ■

Proof of Corollary 1. By Observation 1, any TS-model of $\alpha \wedge \gamma_0$ is an LTL-model too. For the other direction, by Proposition 4, any THT-model of $\alpha \wedge \gamma_0$ is a total model. Since $\alpha \wedge \gamma_0$ has no non-total models, any model $\langle \mathbf{T}, \mathbf{T}\rangle$ is a temporal equilibrium model and so $\mathbf{T}$ is a TS-model. ■

Proof of Lemma 1. For (i) $\Rightarrow$ (ii), suppose (i) holds. As α and β are LTL-equivalent, $\langle \mathbf{H}, \mathbf{T}\rangle \not\models \alpha \to \beta$ amounts to $\langle \mathbf{H}, \mathbf{T}\rangle \models \alpha$ and $\langle \mathbf{H}, \mathbf{T}\rangle \not\models \beta$. Then, it is easy to see that $\langle \mathbf{H}, \mathbf{T}\rangle \models \alpha \wedge \gamma$ because $\langle \mathbf{H}, \mathbf{T}\rangle \not\models \beta$ which is the antecedent of γ. From Proposition 2 (persistence) $\langle \mathbf{T}, \mathbf{T}\rangle \models \alpha \wedge \gamma$ and since α and β have the same total models, $\langle \mathbf{T}, \mathbf{T}\rangle \models \beta \wedge \gamma$ that is, $\mathbf{T} \models \beta \wedge \gamma$ in LTL. But now, as $\beta \wedge (\beta \to \gamma_0)$ is LTL-equivalent to $\beta \wedge \gamma_0$, by Corollary 1, the LTL models of this formula are its TS-models. In particular $\mathbf{T}$ is a TS-model of $\beta \wedge \gamma$. But $\mathbf{T}$ cannot be TS-model of $\alpha \wedge \gamma$ because we had that $\langle \mathbf{H}, \mathbf{T}\rangle$ was a model and $\mathbf{H} < \mathbf{T}$.

For (ii) $\Rightarrow$ (i), suppose (ii) is true. Then, by Observation 1, $\mathbf{T}$ is LTL-model of $\beta \wedge \gamma$, and thus, it is LTL-model of $\alpha \wedge \gamma$ too, since α and β are LTL-equivalent. But as $\mathbf{T}$ is not TS-model of $\alpha \wedge \gamma$, this means there exists some $\mathbf{H} < \mathbf{T}$ such that $\langle \mathbf{H}, \mathbf{T}\rangle \models \alpha \wedge \gamma$ and so $\langle \mathbf{H}, \mathbf{T}\rangle \models \alpha$. On the other hand, $\langle \mathbf{H}, \mathbf{T}\rangle \not\models \beta$ because, otherwise, it would satify $\beta \wedge \gamma$ and, as $\langle \mathbf{H}, \mathbf{T}\rangle$ is non-total, $\mathbf{T}$ could not be TS-model of $\beta \wedge \gamma$. As a result, $\langle \mathbf{H}, \mathbf{T}\rangle \not\models \alpha \to \beta$. ■

Proof of Corollary 2. Suppose $\mathbf{T}$ is a TS-model of $\alpha \wedge \gamma$. By Observation 1, $\mathbf{T}$ is an LTL model of $\alpha \wedge \gamma$ too, and so, it is an LTL model of $\beta \wedge \gamma$, because α and β are LTL-equivalent. But, as discussed in the proof of Theorem 2, any LTL-model of $\beta \wedge \gamma$ is also a TS-model of that theory. ■