

Strongly Equivalent Temporal Logic Programs^{*}

Felicidad Aguado, Pedro Cabalar, Gilberto Pérez, and Concepción Vidal

Dept. Computación,
Corunna University, Spain
{aguado,cabalar,gperez,eicovima}@udc.es

Abstract. This paper analyses the idea of strong equivalence for transition systems represented as logic programs under the Answer Set Programming (ASP) paradigm. To check strong equivalence, we use a linear temporal extension of Equilibrium Logic (a logical characterisation of ASP) and its monotonic basis, the intermediate logic of Here-and-There (HT). Trivially, equivalence in this temporal extension of HT provides a sufficient condition for temporal strong equivalence and, as we show in the paper, it can be transformed into a provability test into the standard Linear Temporal Logic (LTL), something that can be automatically checked using any of the LTL available provers. The paper shows an example of the potential utility of this method by detecting some redundant rules in a simple actions reasoning scenario.

1 Introduction

The paradigm of *Answer set programming* (ASP) [1,2] (based on the *stable models* semantics [3]) constitutes one of the most successful examples of logical non-monotonic formalisms applied to Knowledge Representation [4,5] in Artificial Intelligence. Probably, the reasons for this success are both related to its powerful representational features and, at the same time, to the availability of an increasing number of efficient ASP solvers (see [6]) that allow its application to many real scenarios. Concerning the formalism properties, ASP is characterised by providing nonmonotonic reasoning with a rich and flexible syntax, initially born from logic programming, but continuously extended thereafter along the research history in the area, without overlooking its original semantic simplicity. An important breakthrough in this sense has been the logical characterisation of ASP in terms of *Equilibrium Logic* [7] that has opened, for instance, the study of strong equivalence [8] (the main topic of this paper) and has recently allowed the extension of the stable models semantics for arbitrary first order theories [9,10].

As for the practical applications of ASP, perhaps one of the most outstanding and frequent uses has been the representation and automated reasoning for action domains, solving typical problems like prediction, explanation, planning

^{*} This research is partially supported by Spanish Ministry MEC, research project TIN-15455-C03-02.

or diagnostics. Default negation plays here a crucial role, as it allows representing the rule of *inertia* (that can be stated as “a fluent remains unchanged by default”) and avoid in this way the frame problem [11]. ASP can also be naturally used for solving other typical representational problems in Reasoning about Actions and Change, and is in fact the basis for a family¹ of high level action languages [14]. The use of ASP solvers for action domains, however, has some limitations, as partly explained by the complexity class that these solvers allow to capture (Σ_2^P in the most general case). In practice, this means for instance that, when solving a planning problem (which lies in PSPACE-completeness) we must fix an *a priori* plan length so that a ground logic program can be eventually generated. The search for a minimal plan consists then in gradually incrementing this plan length until a stable model is found. A first obvious drawback of this approach is that it is not possible to establish when a given planning problem has *no solution* of any length at all. A second and more elaborated problem is that it is impossible to establish when two descriptions of the same transition system are *strongly equivalent*, i.e., when they will behave in the same way regardless any additional rules we include and any path length we consider.

At a first sight, it could be thought that this problem for checking strong equivalence of temporal scenarios is only due to the restriction to ground programs. For instance, there already exists a tool (SE-TEST [15]) that allows checking strong equivalence of logic programs with variables, without requiring their previous grounding. However, temporal domains require something else than variables: in order to represent transition rules, we must (at least) be able to refer to a situation index variable I and its successor situation $I + 1$. This forces us either to deal with integer numbers, or at least, with a Peano like representation of the form $s(I)$ with a function symbol s to represent the successor. Unfortunately, SE-TEST is exclusively thought for Datalog programs, i.e. functions or unbounded integers are not allowed.

As we mentioned above, the idea of *strong equivalence* was introduced in [8], where the following question was considered: when can we safely replace a piece of knowledge representation by an “equivalent” one independently of the context? Formally, we say that two logic programs Π_1 and Π_2 are *strongly equivalent* when, for any arbitrary logic program Π , both $\Pi_1 \cup \Pi$ and $\Pi_2 \cup \Pi$ have the same stable models. Note that, for a monotonic logic, this property trivially collapses to regular equivalence (i.e., coincidence of sets of models) of Π_1 and Π_2 . However, when a nonmonotonic entailment is involved, the addition of a set of rules Π may have different effects on the sets of stable models of Π_1 and Π_2 , so that strong equivalence is indeed a *stronger* property than regular equivalence. In [8] it was shown that two logic programs are strongly equivalent (under the stable models semantics) if and only if they are equivalent under the intermediate logic of *Here-and-There* [18], the monotonic basis of Equilibrium Logic.

¹ To be precise, part of this family relies on the formalism of Causal Theories [12] but, as shown in [13], this formalism can be reduced to ASP as well.

In this paper we consider the study of strong equivalence for logic programs that represent transition systems. To this aim, we revisit a temporal extension of Equilibrium Logic proposed in [19] which consists in the inclusion of modal operators as those handled in Linear Temporal Logic (LTL) [20,21]. This extension, called *Temporal Equilibrium Logic*, immediately provides us with a sufficient condition for strong equivalence of temporal logic programs: we can simply check regular equivalence in its monotonic basis, a logic we called *Temporal Here-and-There* (THT) [19]. The main contribution of the paper is the automation of this test for strong equivalence so that, using a similar translation to those presented in [22,23,10], we transform a THT formula into LTL and use an LTL prover afterwards – in particular, we ran our experiments on the *Logics Workbench* [24].

The paper is organised as follows. In the next section, we introduce a simple motivating example, extracted from the Reasoning about Actions literature, to show the kind of problems we are interested in, proposing a pair of strong equivalence tests in this domain. In Section 3 we revisit the syntax and semantics of *Temporal Here-and-There* (THT) and we propose a models selection criterion to define the nonmonotonic *Temporal Equilibrium Logic* (TEL). Section 4 presents the translation from THT into LTL, whereas Section 5 applies this translation to answer the questions proposed in Section 3. Finally, Section 6 contains the conclusions and future work.

2 A Simple Motivating Example

Consider the following simple and well-known scenario [25] from Reasoning about Actions literature.

Example 1. An electric circuit consists of a battery, two switches and a light bulb. The switches are serially connected, as shown in Figure 2. The system state is expressed in terms of three propositional fluents sw_1, sw_2 and $light$, whose negations are represented with a bar on top of each fluent symbol. The state of each switch sw_i can be alternated by performing a corresponding action $toggle_i$. \square

For simplicity, we assume that we do not handle concurrent actions. A possible representation of this scenario as an ASP logic program, Π_1 , is shown below:

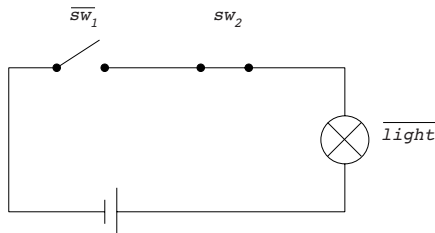


Fig. 1. A simple electric circuit

$$sw_j(I+1) \leftarrow toggle_j(I), \overline{sw}_j(I) \quad (1)$$

$$\overline{sw}_j(I+1) \leftarrow toggle_j(I), sw_j(I) \quad (2)$$

$$\overline{light}(I+1) \leftarrow toggle_j(I), sw_j(I) \quad (3)$$

$$light(I+1) \leftarrow toggle_1(I), \overline{sw}_1(I), sw_2(I) \quad (4)$$

$$light(I+1) \leftarrow toggle_2(I), \overline{sw}_2(I), sw_1(I) \quad (5)$$

$$\perp \leftarrow toggle_1(I), toggle_2(I) \quad (6)$$

$$f(I+1) \leftarrow f(I), not \overline{f}(I+1) \quad (7)$$

$$\overline{f}(I+1) \leftarrow \overline{f}(I), not f(I+1) \quad (8)$$

$$\perp \leftarrow f(I), \overline{f}(I) \quad (9)$$

where $j \in \{1, 2\}$ and $f \in \{sw_1, sw_2, light\}$, and we assume that each symbol like \overline{f} is actually treated as a new predicate. Variable $I = 0 \dots n-1$ represents an integer index for each situation in the temporal narrative. Rules (1)-(5) are the *effect axioms*, capturing all the direct effects of actions. Rule (6) just avoids performing concurrent actions. Rules (7) and (8) represent the inertia default for each fluent f . Finally, (9) expresses that each fluent f and its explicit negation \overline{f} cannot be simultaneously true².

As explained in the introduction, a planning problem would include additional facts (representing the initial state) and rules for generating actions and expressing a plan goal. But the most important additional step is deciding a finite limit n for variable I , so that the above program can be grounded.

Assume now that we want to modify Π_1 after noticing that the truth value of *light* is actually determined by the value of the two switches, becoming in this way an indirect effect or *ramification*³. In other words, we consider the addition of rules:

$$light(I) \leftarrow sw_1(I), sw_2(I) \quad (10)$$

$$\overline{light}(I) \leftarrow \overline{sw}_1(I) \quad (11)$$

$$\overline{light}(I) \leftarrow \overline{sw}_2(I) \quad (12)$$

If we call $\Pi_2 = \Pi_1 \cup \{(10) - (12)\}$, we may reasonably propose the following questions:

(Q1) Can we safely remove now from Π_2 the effect axioms for fluent *light* (3)-(5)?

(Q2) Assume we removed (3)-(5). Since *light* is “defined” now in terms of sw_1 and sw_2 , can we safely remove from Π_2 the inertia rules for *light*?

In order to answer these questions, the current tools for testing strong equivalence for ground programs [26,27] cannot provide successful answers in an automated way, as they *need* a numeric value for the path length n to be previously fixed.

² Although perhaps not needed in this particular case, this axiom must be part of any general translation of an action theory. It is not difficult to see how the above scenario can be easily modified to introduce new contradictory effects that would not be detected in absence of (9).

³ In fact, this example were used in [25] to illustrate possible representational problems for a suitable treatment of action ramifications.

3 Linear Temporal Here-and-There (THT)

We proceed now to recall the main definitions of the temporal extension of Equilibrium Logic, beginning with its monotonic basis. The logic of *Linear Temporal Here-and-There* (THT) was defined as follows. We start from a finite set of atoms Σ called the *propositional signature*. A (temporal) *formula* is defined as any combination of the atoms in Σ with the classical connectives $\wedge, \vee, \rightarrow, \perp$, the (unary) temporal operators \Box (to be read as “always” or “from now on”), \Diamond (“eventually”), \circ (“next”) and the (binary) temporal operators \mathcal{U} (“until”), \mathcal{W} (“weak until”) and \mathcal{B} (“before”). Negation is defined as $\neg\varphi \stackrel{def}{=} \varphi \rightarrow \perp$ whereas $\top \stackrel{def}{=} \neg\perp$. As usual, $\varphi \leftrightarrow \psi$ stands for $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$. We also allow the abbreviation $\circ^i\varphi \stackrel{def}{=} \circ(\circ^{i-1}\varphi)$ for $i > 0$ and $\circ^0\varphi \stackrel{def}{=} \varphi$.

A (temporal) *interpretation* M is an infinite sequence of pairs $\langle H_i, T_i \rangle$ with $i = 0, 1, 2, \dots$ where $H_i \subseteq T_i$ are sets of atoms. From an ASP point of view and informally speaking, atoms in T_i would play the role of an interpretation (that depends on a time parameter i) we would use for building a program reduct, whereas atoms in H_i will be minimised and later required to coincide with T_i . For simplicity, given a temporal interpretation, we write \overline{H} (resp. \overline{T}) to denote the sequence of pair components H_0, H_1, \dots (resp. T_0, T_1, \dots). Using this notation, we will sometimes abbreviate the interpretation as $M = \langle \overline{H}, \overline{T} \rangle$. An interpretation $M = \langle \overline{H}, \overline{T} \rangle$ is said to be total when $\overline{H} = \overline{T}$. We say that an interpretation $M = \langle \overline{H}, \overline{T} \rangle$ satisfies a formula φ at some sequence index i , written $M, i \models \varphi$, when any of the following hold:

1. $M, i \models p$ if $p \in H_i$, for any atom p .
2. $M, i \models \varphi \wedge \psi$ if $M, i \models \varphi$ and $M, i \models \psi$.
3. $M, i \models \varphi \vee \psi$ if $M, i \models \varphi$ or $M, i \models \psi$.
4. $\langle \overline{H}, \overline{T} \rangle, i \models \varphi \rightarrow \psi$ if both:
 - (a) $\langle \overline{H}, \overline{T} \rangle, i \not\models \varphi$ or $\langle \overline{H}, \overline{T} \rangle, i \models \psi$;
 - (b) $\langle \overline{T}, \overline{T} \rangle, i \not\models \varphi$ or $\langle \overline{T}, \overline{T} \rangle, i \models \psi$.
5. $M, i \models \circ\varphi$ if $M, i+1 \models \varphi$.
6. $M, i \models \Box\varphi$ if for all $j \geq i$, $M, j \models \varphi$.
7. $M, i \models \Diamond\varphi$ if there exists some $j \geq i$, $M, j \models \varphi$.
8. $M, i \models \varphi \mathcal{U} \psi$ if there exists $j \geq i$, $M, j \models \psi$ and $M, k \models \varphi$ for all k such that $i \leq k < j$.
9. $M, i \models \varphi \mathcal{W} \psi$ if either $M, i \models \varphi \mathcal{U} \psi$ or, for all $j \geq i$, $M, j \models \varphi$.
10. $M, i \models \varphi \mathcal{B} \psi$ if for all $j \geq i$, either $M, j \models \psi$ or there exists some k , $i \leq k < j$ such that $M, k \models \varphi$.

We assume that a finite sequence $\langle H_0, T_0 \rangle \dots \langle H_n, T_n \rangle$ with $n \geq 0$ is an abbreviation of the infinite sequence $\langle \overline{H}', \overline{T}' \rangle$ with $H'_i = H_i$, $T'_i = T_i$ for $i = 0, \dots, n$ and $H'_i = H_n$, $T'_i = T_n$ for $i > n$.

The logic of THT is an orthogonal combination of the logic of HT and the (standard) linear temporal logic (LTL) [21]. When we restrict temporal interpretations to finite sequences of length 1, that is $\langle H_0, T_0 \rangle$ and disregard temporal

operators, we obtain the logic of HT. On the other hand, if we restrict the semantics to total interpretations, $\langle \overline{T}, \overline{T} \rangle, i \models \varphi$ corresponds to satisfaction of formulas $\overline{T}, i \models \varphi$ in LTL.

A *theory* is any set of formulas. An interpretation M is a *model* of a theory Γ , written $M \models \Gamma$, if $M, 0 \models \alpha$, for each formula $\alpha \in \Gamma$. A formula φ is *valid* if $M, 0 \models \varphi$ for any M . The following are valid formulas in THT (and in LTL too):

$$\diamond \varphi \leftrightarrow \top \mathcal{U} \varphi \quad (13)$$

$$\square \varphi \leftrightarrow \perp \mathcal{B} \varphi \quad (14)$$

$$\varphi \mathcal{W} \psi \leftrightarrow \varphi \mathcal{U} \psi \vee \square \varphi \quad (15)$$

$$\varphi \mathcal{U} \psi \leftrightarrow \varphi \mathcal{W} \psi \wedge \diamond \psi \quad (16)$$

$$\neg(\varphi \mathcal{U} \psi) \leftrightarrow \neg \varphi \mathcal{B} \neg \psi \quad (17)$$

$$\neg(\varphi \mathcal{B} \psi) \leftrightarrow \neg \varphi \mathcal{U} \neg \psi \quad (18)$$

$$\neg \square \varphi \leftrightarrow \diamond \neg \varphi \quad (19)$$

$$\neg \diamond \varphi \leftrightarrow \square \neg \varphi \quad (20)$$

$$\bigcirc \neg \varphi \leftrightarrow \neg \bigcirc \varphi \quad (21)$$

$$\bigcirc(\varphi \wedge \psi) \leftrightarrow \bigcirc \varphi \wedge \bigcirc \psi \quad (22)$$

$$\bigcirc(\varphi \vee \psi) \leftrightarrow \bigcirc \varphi \vee \bigcirc \psi \quad (23)$$

$$\bigcirc(\varphi \rightarrow \psi) \leftrightarrow (\bigcirc \varphi \rightarrow \bigcirc \psi) \quad (24)$$

$$\varphi \mathcal{U} \psi \leftrightarrow \psi \vee (\varphi \wedge \bigcirc(\varphi \mathcal{U} \psi)) \quad (25)$$

$$\varphi \mathcal{B} \psi \leftrightarrow \psi \wedge (\varphi \vee \bigcirc(\varphi \mathcal{B} \psi)) \quad (26)$$

Theorems (13)-(15) allow defining \square , \diamond and \mathcal{W} in terms of \mathcal{U} and \mathcal{B} . The formulas (17) and (18) correspond to the De Morgan axioms between operators \mathcal{U} and \mathcal{B} . It is easy to see that, together with (13) and (14) they directly imply the corresponding De Morgan axioms (19) and (20) for \square and \diamond . An important difference with respect to LTL is that, when using these De Morgan axioms, some care must be taken if double negation is involved. For instance, by (19), the formula $\neg \diamond \neg \varphi$ is equivalent to $\neg \neg \square \varphi$, but this is not in general equivalent to $\square \varphi$. A simple counterexample is the interpretation $\langle \overline{H}, \overline{T} \rangle$ with all $T_i = \{p\}$ but some $H_j = \emptyset$, as it satisfies $\neg \neg \square p$ but not $\square p$. As a result, we cannot further define \mathcal{B} (resp. \square) in terms of \mathcal{U} (resp. \diamond) or vice versa, as happens in LTL. We have included other LTL standard properties like (21)-(24) to show that the “shifting” behaviour of \bigcirc with respect to classical connectives is the same as in LTL, or (25) and (26) that represent the inductive propagation of \mathcal{U} and \mathcal{B} respectively.

The (*Linear*) *Temporal Equilibrium Logic* (TEL) is a nonmonotonic version of THT where we establish a models selection criterion. Given two interpretations $M = \langle \overline{H}, \overline{T} \rangle$ and $M' = \langle \overline{H}', \overline{T}' \rangle$ we say that M is *lower than* M' , written $M \leq M'$, when $\overline{T} = \overline{T}'$ and for all $i \geq 0$, $H_i \subseteq H'_i$. As usual, $M < M'$ stands for $M \leq M'$ but $M \neq M'$.

Definition 1 (Temporal Equilibrium Model). *An interpretation M is a temporal equilibrium model of a theory Γ if M is a total model of Γ and there is no other model $M' < M$ of Γ . \square*

Note that any temporal equilibrium model is total, that is, it has the form $\langle \overline{T}, \overline{T} \rangle$ and so can be actually seen as an interpretation \overline{T} in the standard LTL.

The idea behind temporal equilibrium models is that they capture the answer sets of programs that depend on a fixed temporal index. This is expressed by the following proposition extracted from [19].

Proposition 1. *Let Γ be a combination of non-modal connectives $\wedge, \vee, \neg, \rightarrow, \perp$ with expressions like $\bigcirc^i p$, being p an atom, and let n be the maximum value for i in all $\bigcirc^i p$ occurring in Γ . Then $\langle \overline{T}, \overline{T} \rangle$ is a temporal equilibrium model of Γ iff (1) $T_i = \emptyset$ for all $i > n$; and (2) $\langle X, X \rangle$ with $X = \bigcup_{i=0}^n \{\bigcirc^i p \mid p \in T_i\}$ is an equilibrium model of Γ , reading each ' $\bigcirc^i p$ ' as a new atom in the signature. \square*

That is, once $\square, \diamond, \mathcal{U}$ and \mathcal{W} are removed, we can reduce temporal equilibrium models to (non-temporal) equilibrium models (that is, answer sets) for an extended signature with atoms like $\bigcirc^i p$. To illustrate the effect of this definition, let us consider a pair of examples from [19]. Take, for instance, the theory $\{\diamond p\}$. This would correspond to an infinite disjunction of the form $p \vee \bigcirc p \vee \bigcirc \bigcirc p \vee \dots$ or, when making time explicit, to an ASP disjunctive rule like $p(0) \vee p(1) \vee p(2) \vee \dots$. This generates temporal equilibrium models where p is false in every situation excepting for exactly one index i where p is made true. In other words, the temporal equilibrium models of theory $\{\diamond p\}$ are captured by the LTL models of the formula $\neg p \ \mathcal{U} \ (p \wedge \bigcirc \square \neg p)$.

As a second example, take the theory Γ just consisting of the formula $\square(\neg p \rightarrow \bigcirc p)$. This would informally correspond to an infinite program with the rules $p(1) \leftarrow \neg p(0)$; $p(2) \leftarrow \neg p(1)$; $p(3) \leftarrow \neg p(2)$; etc. It is not difficult to see that the temporal equilibrium models of Γ are captured by the LTL models of $\neg p \wedge \square(\neg p \leftrightarrow \bigcirc p)$.

In [19] it was shown as an example how TEL can be used to encode the action language \mathcal{B} [14] in a straightforward way.

The definition of strong equivalence for TEL theories is extended from the non-temporal case in an obvious way. An important observation is that proving the equivalence of two temporal theories in THT is a trivial sufficient condition for their strong equivalence under TEL – if they have the same THT models, they will always lead to the same set of temporal equilibrium models, regardless the context. The next section shows how this THT equivalence test can be transformed into provability in standard LTL.

4 Translating THT into LTL

Given a propositional signature Σ , let us denote $\Sigma^* = \Sigma \cup \{p' \mid p \in \Sigma\}$ which is going to be the new propositional signature in LTL. For any temporal formula φ we define its translation φ^* as follows:

1. $\perp^* \stackrel{def}{=} \perp$
2. $p^* \stackrel{def}{=} p'$ for any $p \in \Sigma$
3. $(\odot\varphi)^* \stackrel{def}{=} \odot\varphi^*$, if $\odot \in \{\square, \diamond, \bigcirc\}$
4. $(\varphi \odot \psi)^* \stackrel{def}{=} \varphi^* \odot \psi^*$, when $\odot \in \{\wedge, \vee, \mathcal{U}, \mathcal{W}, \mathcal{B}\}$
5. $(\varphi \rightarrow \psi)^* \stackrel{def}{=} (\varphi \rightarrow \psi) \wedge (\varphi^* \rightarrow \psi^*)$

From the last point and the fact that $\neg\varphi = \varphi \rightarrow \perp$, it follows that $(\neg\varphi)^* = (\varphi \rightarrow \perp) \wedge (\varphi^* \rightarrow \perp) = \neg\varphi \wedge \neg\varphi^*$. Similarly, $(\varphi \leftrightarrow \psi)^* = (\varphi \leftrightarrow \psi) \wedge (\varphi^* \leftrightarrow \psi^*)$.

We associate to any THT interpretation $M = \langle \overline{H}, \overline{T} \rangle$ the LTL interpretation $M^t = \overline{T}$ in LTL defined as the sequence of sets of atoms $I_i = \{p' \mid p \in H_i\} \cup T_i$, for any $i \geq 0$. Informally speaking, M^t considers a new primed atom p' per each $p \in \Sigma$ in the original signature. In the LTL interpretation, the primed atom p' represents the fact that p occurs at some point in the \overline{H} component, whereas the original symbol p is used to represent an atom in \overline{T} . As a THT interpretation must satisfy $H_i \subseteq T_i$ by construction, we may have LTL interpretations that do not correspond to any THT one. In particular, for an arbitrary \overline{T} , we will only be able to form some M such that $M^t = \overline{T}$ when the set of primed atoms at each I_i is a subset of the non-primed ones. In other words, only LTL interpretations \overline{T} satisfying the axiom:

$$\square(p' \rightarrow p) \tag{27}$$

will have a corresponding THT interpretation M such that $\overline{T} = M^t$.

Example 2. $M = ((\emptyset, \{p, q\}), (\{p\}, \{p, q\}), (\{q\}, \{q\}))$ is a model of the theory $\{\square(\neg p \rightarrow q) \wedge \diamond q\}$. In the same way, the corresponding interpretation $M^t = (\{p, q\}, \{p', p, q\}, \{q', q\})$ is a model of

$$\begin{aligned} &(\square(\neg p \rightarrow q) \wedge \diamond q)^* && \leftrightarrow \square(\neg p \rightarrow q)^* \wedge (\diamond q)^* && \leftrightarrow \\ &\square((\neg p \rightarrow q) \wedge ((\neg p)^* \rightarrow q')) \wedge \diamond q' && \leftrightarrow \square((\neg p \rightarrow q) \wedge (\neg p \rightarrow q')) \wedge \diamond q'. \end{aligned}$$

In general:

Theorem 1. *Let $M = \langle \overline{H}, \overline{T} \rangle$ be any THT interpretation and φ any formula. For any $i \geq 0$, it holds that*

- (a) $\langle \overline{H}, \overline{T} \rangle, i \models \varphi$ if and only if $M^t, i \models \varphi^*$ in LTL; and
- (b) $\langle \overline{T}, \overline{T} \rangle, i \models \varphi$ if and only if $M^t, i \models \varphi$ in LTL.

Proof. We proceed by induction. For the base case, it trivially holds for \perp whereas for an atom p , we have these equivalence chains

- (a) $\langle \overline{H}, \overline{T} \rangle, i \models p \Leftrightarrow (p \in H_i) \Leftrightarrow (p' \in I_i) \Leftrightarrow (M^t, i \models p')$.
- (b) $\langle \overline{T}, \overline{T} \rangle, i \models p \Leftrightarrow (p \in T_i) \Leftrightarrow (p \in I_i) \Leftrightarrow (M^t, i \models p)$

For the inductive step, we detail the proof for the classical connective \rightarrow , the (unary) temporal operator \bigcirc and the (binary) temporal operator \mathcal{U} . For the classical connectives \wedge and \vee the proof is straightforward; for connective \mathcal{B} , it is completely analogous to that for \mathcal{U} ; and finally, the rest of connectives can be defined in terms of the previous ones.

1. To prove (a) for the implication we have the chain of equivalent conditions:

$$\begin{aligned} \langle \overline{H}, \overline{T} \rangle, i \models \varphi \rightarrow \psi &\stackrel{def}{\iff} \left\{ \begin{array}{l} \langle \overline{H}, \overline{T} \rangle, i \not\models \varphi \text{ or } \langle \overline{H}, \overline{T} \rangle, i \models \psi \\ \text{and} \\ \langle \overline{T}, \overline{T} \rangle, i \not\models \varphi \text{ or } \langle \overline{T}, \overline{T} \rangle, i \models \psi \end{array} \right\} \stackrel{ind}{\iff} \\ \left\{ \begin{array}{l} M^t, i \not\models \varphi^* \text{ or } M^t, i \models \psi^* \\ \text{and} \\ M^t, i \not\models \varphi \text{ or } M^t, i \models \psi \end{array} \right\} &\iff \left\{ \begin{array}{l} M^t, i \models (\varphi^* \rightarrow \psi^*) \\ \text{and} \\ M^t, i \models (\varphi \rightarrow \psi) \end{array} \right\} \stackrel{def}{\iff} \\ M^t, i \models (\varphi \rightarrow \psi)^* & \end{aligned}$$

For proving (b) it suffices with considering, in each of the three pairs of conjunctive conditions above, only the second conjunct.

2. For operator \bigcirc , note that

$$\begin{aligned} \text{(a)} \quad \langle \overline{H}, \overline{T} \rangle, i \models \bigcirc \varphi &\stackrel{def}{\iff} \langle \overline{H}, \overline{T} \rangle, i+1 \models \varphi \stackrel{ind}{\iff} M^t, i+1 \models \varphi^* \stackrel{def}{\iff} \\ M^t, i \models \bigcirc \varphi^* & \\ \text{(b)} \quad \langle \overline{T}, \overline{T} \rangle, i \models \bigcirc \varphi &\stackrel{def}{\iff} \langle \overline{T}, \overline{T} \rangle, i+1 \models \varphi \stackrel{ind}{\iff} M^t, i+1 \models \varphi \stackrel{def}{\iff} \\ M^t, i \models \bigcirc \varphi & \end{aligned}$$

3. Finally, for \mathcal{U} it follows that:

$$\begin{aligned} \text{(a)} \quad \langle \overline{H}, \overline{T} \rangle, i \models \varphi \mathcal{U} \psi &\stackrel{def}{\iff} \\ \exists j \geq i, (M, j \models \psi) \text{ and } \forall k \text{ s.t. } i \leq k < j, (M, k \models \varphi) &\stackrel{ind}{\iff} \\ \exists j \geq i, (M^t, j \models \psi^*) \text{ and } \forall k \text{ s.t. } i \leq k < j, (M, k \models \varphi^*) &\stackrel{def}{\iff} \\ M^t, i \models \varphi^* \mathcal{U} \psi^* &\stackrel{def}{\iff} M^t, i \models (\varphi \mathcal{U} \psi)^* \\ \text{(b)} \quad \langle \overline{T}, \overline{T} \rangle, i \models \varphi \mathcal{U} \psi &\stackrel{def}{\iff} \\ \exists j \geq i, (M, j \models \psi) \text{ and } \forall k \text{ s.t. } i \leq k < j, (M, k \models \varphi) &\stackrel{ind}{\iff} \\ \exists j \geq i, (M^t, j \models \psi) \text{ and } \forall k \text{ s.t. } i \leq k < j, (M, k \models \varphi) &\stackrel{def}{\iff} \\ M^t, i \models \varphi \mathcal{U} \psi & \end{aligned} \quad \square$$

Theorem 2 (Main theorem). *Let Γ_1 and Γ_2 be a pair of temporal theories, and $\bigwedge \Gamma_1$ and $\bigwedge \Gamma_2$ the conjunctions of their respective sets of formulas. Then Γ_1 and Γ_2 are strongly equivalent with respect to temporal equilibrium models if the formula (27) $\rightarrow (\bigwedge \Gamma_1 \leftrightarrow \bigwedge \Gamma_2)^*$ is valid in LTL.*

Proof. Once the axiom schemata (27) are fixed as hypotheses, Theorem 1 allows us to establish a one to one correspondence between models of $\bigwedge \Gamma_1 \leftrightarrow \bigwedge \Gamma_2$ in THT and models of $(\bigwedge \Gamma_1 \leftrightarrow \bigwedge \Gamma_2)^*$ in LTL. Thus, if $(\bigwedge \Gamma_1 \leftrightarrow \bigwedge \Gamma_2)^*$ is LTL valid, this means that $\bigwedge \Gamma_1$ and $\bigwedge \Gamma_2$ are THT-equivalent, which is a sufficient condition for strong equivalence in TEL. \square

As we will show in the next section, we may also use this result to detect a redundant formula φ in some theory Γ . To this aim, we would have to show that Γ and $\Gamma' = \Gamma \setminus \{\varphi\}$ are strongly equivalent. From the theorem above, it follows that:

Corollary 1. *Let Γ be a temporal theory, $\bigwedge \Gamma$ the conjunction of its formulas and φ some arbitrary temporal formula. Then Γ and $\Gamma \cup \{\varphi\}$ are strongly equivalent if the formula (27) $\rightarrow (\bigwedge \Gamma \rightarrow \varphi)^*$ is valid in LTL.* \square

5 Back to the Example

When representing Example 1 in TEL, the encoding is quite obvious. As happens in Equilibrium Logic, the logic programming notation is directly replaced by a logical syntax, so that the rule arrow ‘ \leftarrow ’, the comma ‘ $,$ ’ and the default negation ‘*not*’ respectively represent standard implication, conjunction and negation. On the other hand, in order to capture the transition rules of program Π_1 , only a small subset of the linear temporal syntax is actually required. All the rules depending on the temporal (universally quantified) variable I will be scoped now by a \square operator, whereas the atoms with argument $I + 1$ will be further preceded by a \bigcirc connective. As a result, we obtain the theory Γ_1 consisting of the formulas:

$$\square(\text{toggle}_j \wedge \overline{sw}_j \rightarrow \bigcirc sw_j) \quad (28)$$

$$\square(\text{toggle}_j \wedge sw_j \rightarrow \bigcirc \overline{sw}_j) \quad (29)$$

$$\square(\text{toggle}_j \wedge sw_j \rightarrow \bigcirc \overline{\text{light}}) \quad (30)$$

$$\square(\text{toggle}_1 \wedge \overline{sw}_1 \wedge sw_2 \rightarrow \bigcirc \text{light}) \quad (31)$$

$$\square(\text{toggle}_2 \wedge \overline{sw}_2 \wedge sw_1 \rightarrow \bigcirc \text{light}) \quad (32)$$

$$\square(\text{toggle}_1 \wedge \text{toggle}_2 \rightarrow \perp) \quad (33)$$

$$\square(f \wedge \neg \bigcirc \overline{f} \rightarrow \bigcirc f) \quad (34)$$

$$\square(\overline{f} \wedge \neg \bigcirc f \rightarrow \bigcirc \overline{f}) \quad (35)$$

$$\square(f \wedge \overline{f} \rightarrow \perp) \quad (36)$$

that respectively correspond to the rules (1)-(9).

In order to answer questions Q1 and Q2 proposed in Section 2, we have implemented the φ^* translation⁴ and fed the LWB linear temporal logic module with the resulting formulas extracted from Corollary 1 to detect redundant rules. In particular, regarding question Q1, we have been able to prove that, given the theory Γ_2 consisting of Γ_1 plus:

$$\square(sw_1 \wedge sw_2 \rightarrow \text{light}) \quad (37)$$

$$\square(\overline{sw}_1 \rightarrow \overline{\text{light}}) \quad (38)$$

$$\square(\overline{sw}_2 \rightarrow \overline{\text{light}}) \quad (39)$$

(that captures *light* as an indirect effect) then rule (30), that specified when *light* became off depending on each *toggle_j*, becomes redundant and can be safely removed from Γ_2 . In fact, we could just prove that (30) followed from the formulas (29), (38) and (39).

However, when we consider the effect axioms (31),(32) we used to specify when the *light* becomes on, our checker just provides a negative answer. Since we are just dealing with a sufficient condition for strong equivalence, this does not provide any concluding information about Q1 for these rules. Fortunately, in this case it is not difficult to see that, in fact, these rules cannot be removed

⁴ Available at <http://www.dc.fi.udc.es/~cabalar/eqwb.html>

in any context without changing the general behaviour of the program. To see why, just think about adding, as a new effect of $toggle_1$, that switch sw_2 is disconnected:

$$\Box(toggle_1 \wedge sw_2 \rightarrow \overline{sw_2})$$

The addition of this new rule would clearly make a different effect depending on (31) is present in the theory or not. In particular, when we perform $toggle_1$ in a situation in which $\overline{sw_1} \wedge sw_2$, if we do not have (31) the light will just remain off, whereas if this formula is included, we will get no equilibrium model.

In order to answer question Q2, we considered theory $T_3 = T_2 \setminus \{(3)-(5)\}$ and tried to see whether (34),(35) were redundant, for the case $f = light$. Although it could seem that $light$ value is “defined” in terms of sw_1 and sw_2 by rules (37)-(39), we actually obtained a negative answer, so no conclusion is obtained. It is easy to see that, in this case, we can find situations in which removing inertia may cause different effects depending on the context. For instance, if we had $light$ and do not provide information for sw_1 and sw_2 or their explicit negations in a given state, the inertia would maintain $light$ in the next situation (toggling yields no effect). If we remove inertia, however, neither $light$ nor \overline{light} would hold in the next state.

If we look for a stronger relation between $light$ and sw_1, sw_2 we can move to consider T_3 , consisting of T_2 and the rules:

$$\Box(light \rightarrow sw_1) \quad \Box(light \rightarrow sw_2) \quad \Box(\neg light \rightarrow \overline{light})$$

we finally obtain that inertia (34),(35) for $f = light$ is redundant. The explanation for this is simple – note that these rules, together with (37) allow concluding (among other things) that $\Box(light \leftrightarrow sw_1 \wedge sw_2)$ which is a double implication, actually defining $light$ as the conjunction of sw_1 and sw_2 in any situation.

6 Conclusions

In this paper we have studied the problem of strong equivalence of logic programs that represent transition systems. To this aim, we have incorporated the set of modal operators typically used in Linear Temporal Logic (LTL) into logic programs, and applied a previously introduced [19] general semantics for any arbitrary (propositional) temporal theory. This semantics is a temporal extension of Equilibrium Logic [7] (a logical characterisation of stable models) and its monotonic basis, the logic of Here-and-There (HT). As a result, we obtained a sufficient condition for checking strong equivalence of temporal logic programs and we implemented a method for reducing this condition to provability in standard LTL. The advantage of this process is that, to the best of our knowledge, it constitutes the first automated tool for guaranteeing strong equivalence of logic programs representing transition systems, as the previous existing checkers either require fixing a numerical path length to obtain a ground program, or in the case of programs with variables [15], cannot properly deal with transition

systems. There exists, however, a very recent prover (QHT [16]) for strong equivalence of programs with variables, which relies on a method proposed in [17] for reducing Quantified Here and There to first order classical logic, allowing in this way, the use of a classical theorem prover as a backend. Unlike [15], QHT allows dealing with functions and so, could potentially handle situation indices with a Peano-like successor function. We leave for future work checking this alternative, although in a general case, the currently presented reduction to LTL has the obvious advantage of handling a decidable inference method.

Of course, the main current drawback of our strong equivalence test is that, when it provides a negative answer, no conclusion can be drawn, since we have not proved yet whether it also constitutes a necessary condition for strong equivalence, something that was obtained [8] for the non-modal case with HT-equivalence. The desirable situation would be, instead, that a negative answer came with a counterexample, that is, a piece of (temporal) program that added to the original ones to be tested yields different effects in each case. This point is left for the immediate future work.

Acknowledgements. We wish to thank David Pearce, Agustín Valverde, Manuel Ojeda, Alfredo Burrieza for their helpful discussions about this work. We are also thankful to the anonymous referees for their comments and observations.

References

1. Marek, V., Truszczyński, M.: Stable models and an alternative logic programming paradigm. In: *The Logic Programming Paradigm: a 25-Year Perspective*, pp. 169–181. Springer, Heidelberg (1999)
2. Niemelä, I.: Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence* 25, 241–273 (1999)
3. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R.A., Bowen, K.A. (eds.) *Logic Programming: Proc. of the Fifth International Conference and Symposium*, vol. 2, pp. 1070–1080. MIT Press, Cambridge (1988)
4. Baral, C.: *Knowledge Representation, Reasoning and Declarative Problem Solving*, pp. 285–316. Elsevier, Amsterdam (2003)
5. Gelfond, M.: Answer Sets. In: *Handbook of Knowledge Representation*, pp. 285–316. Elsevier, Amsterdam (2007)
6. WASP: ASP solvers web page (last update, 2005), <http://dit.unitn.it/~wasp/Solvers/index.html>
7. Pearce, D.: A new logical characterisation of stable models and answer sets. In: Dix, J., Przymusiński, T.C., Moniz Pereira, L. (eds.) *NMELP 1996*. LNCS, vol. 1216. Springer, Heidelberg (1997)
8. Lifschitz, V., Pearce, D., Valverde, A.: Strongly equivalent logic programs. *Computational Logic* 2(4), 526–541 (2001)
9. Pearce, D., Valverde, A.: Towards a first order equilibrium logic for nonmonotonic reasoning. In: Alferes, J.J., Leite, J.A. (eds.) *JELIA 2004*. LNCS (LNAI), vol. 3229, pp. 147–160. Springer, Heidelberg (2004)

10. Ferraris, P., Lee, J., Lifschitz, V.: A new perspective on stable models. In: Proc. of the International Joint Conference on Artificial Intelligence (IJCAI 2007), pp. 372–379 (2004)
11. McCarthy, J., Hayes, P.: Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence Journal* 4, 463–512 (1969)
12. Giunchiglia, E., Lee, J., Lifschitz, V., McCain, N., Turner, H.: Nonmonotonic causal theories. *Artificial Intelligence Journal* 153, 49–104 (2004)
13. Ferraris, P.: Causal theories as logic programs. In: Proc. of the 20th Workshop on Logic Programming (WLP 2006) (2006)
14. Gelfond, M., Lifschitz, V.: Action languages. *Linköping Electronic Articles in Computer and Information Science* 3(16) (1998)
15. Eiter, T., Faber, W., Traxler, P.: Testing strong equivalence of datalog programs - implementation and examples. In: Baral, C., Greco, G., Leone, N., Terracina, G. (eds.) LPNMR 2005. LNCS (LNAI), vol. 3662, pp. 437–441. Springer, Heidelberg (2005)
16. Cabalar, P.: QHT - a prover for quantified here and there (2008), <http://www.dc.fi.udc.es/~cabalar/eqwb.html>
17. Lifschitz, V., Pearce, D., Valverde, A.: A characterization of strong equivalence for logic programs with variables. In: Baral, C., Brewka, G., Schlipf, J. (eds.) LPNMR 2007. LNCS (LNAI), vol. 4483. Springer, Heidelberg (2007)
18. Heyting, A.: Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften. Physikalisch-mathematische Klasse*, 42–56 (1930)
19. Cabalar, P., Vega, G.P.: Temporal equilibrium logic: a first approach. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2007. LNCS, vol. 4739, pp. 241–248. Springer, Heidelberg (2007)
20. Kamp, J.A.: Tense Logic and the Theory of Linear Order. PhD thesis, University of California at Los Angeles (1968)
21. Manna, Z., Pnueli, A.: *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer, Heidelberg (1991)
22. Pearce, D., Tompits, H., Woltran, S.: Encodings of equilibrium logic and logic programs with nested expressions. In: Bradzil, P., Jorge, A. (eds.) EPIA 2001. LNCS (LNAI), vol. 2258, pp. 306–320. Springer, Heidelberg (2001)
23. Lin, F.: Reducing strong equivalence of logic programs to entailment in classical propositional logic. In: Bradzil, P., Jorge, A. (eds.) Proc. of the 8th Intl. Conf. on Principles and Knowledge Representation and Reasoning (KR 2002), pp. 170–176. Morgan Kaufmann, San Francisco (2002)
24. Heurding, A., Jäger, G., Schwendimann, S., Seyfried, M.: A logics workbench. *AI Communications* 9(2), 53–58 (1996)
25. Thielscher, M.: Ramification and causality. *Artificial Intelligence Journal* 89(1–2), 317–364 (1997)
26. Valverde, A.: tabeql: A tableau based suite for equilibrium logic. In: Alferes, J.J., Leite, J.A. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 734–737. Springer, Heidelberg (2004)
27. Chen, Y., Lin, F., Li, L.: SELP - a system for studying strong equivalence between logic programs. In: Baral, C., Greco, G., Leone, N., Terracina, G. (eds.) LPNMR 2005. LNCS (LNAI), vol. 3662, pp. 442–446. Springer, Heidelberg (2005)