Stable Models for Temporal Theories*

Invited

Pedro Cabalar

Department of Computer Science,
University of Corunna, Spain.
`cabalar@udc.es`

Abstract. This work makes an overview on an hybrid formalism that combines the syntax of Linear-time Temporal Logic (LTL) with a non-monotonic selection of models based on Equilibrium Logic. The resulting approach, called Temporal Equilibrium Logic, extends the concept of a stable model for any arbitrary modal temporal theory, constituting a suitable formal framework for the specification and verification of dynamic scenarios in Answer Set Programming (ASP). We will recall the basic definitions of this logic and explain their effects on some simple examples. After that, we will proceed to summarize the advances made so far, both in the fundamental realm and in the construction of reasoning tools. Finally, we will explain some open topics, many of them currently under study, and foresee potential challenges for future research.

1 Introduction

The birth of Non-Monotonic Reasoning (NMR) in the 1980's was intimately related to temporal reasoning in action domains. The solution to the *frame problem* [1] (the unfeasibility of explicitly specifying all the non-effects of an action) played a central role in research on NMR formalisms capable of representing defaults. In particular, the area of reasoning about actions and change was initially focused on properly capturing the *inertia law*, a dynamic default which can be phrased as "fluent values remain unchanged along time, unless there is evidence on the contrary." NMR was also essential to deal with other typical representational problems in action theories, such as the *ramification* and the *qualification* problems.

The combination of temporal reasoning and NMR in action theories was typically done inside the realm of first order logic. Classical action languages such as *Situation Calculus* [1] or *Event Calculus* [2] have combined some NMR technique, usually predicate circumscription [3], with a first-order formalisation of time using temporal predicates and objects (situations or events, respectively). In this way, we get very rich and expressive formalisms without limitations on the quantification of temporal terms or the construction of arbitrary expressions

* This research was partially supported by Spanish MEC project TIN2013-42149-P and Xunta de Galicia grant GPC 2013/070.

involving them, although we inherit the undecidability of first order logic in the general case.

Another way of dealing with temporal reasoning in NMR approaches has been the use of modal temporal logic, a combination perhaps less popular[1], but not unfrequent in the literature [5,6,7]. 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 [8] for reasoning about actions using Logic Programming (LP), and in the family of action languages [9] inspired on that methodology.

With the consolidation of Answer Set Programming (ASP) [10,11] as a successful paradigm for practical NMR, many examples and benchmarks formalising dynamic scenarios became available. ASP inherited the treatment of time as an integer index from LP-based action languages but, in practice, it further restricted all reasoning tasks to finite narratives, something required for grounding time-related variables. To illustrate this orientation, 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 arbitrarily fix p true at any moment and that that either p or q holds initially. A typical ASP representation of this problem could look like this:

$$p(0) \vee q(0) \tag{1}$$

$$p(I{+}1) \leftarrow p(I), \mathit{not}\ q(I{+}1), \mathit{sit}(I) \tag{2}$$

$$q(I{+}1) \leftarrow q(I), \mathit{not}\ p(I{+}1), \mathit{sit}(I) \tag{3}$$

$$p(I) \vee \mathit{not}\ p(I) \leftarrow \mathit{sit}(I) \tag{4}$$

where (1) describes the initial state, (2) and (3) are the inertia rules for p and q, and (4) acts as a choice rule[2] allowing the introduction of p at any situation. Predicate sit would have some finite domain $0 \dots n$ for some constant $n \geq 0$. A planning problem can be solved incrementally [12], using an iterative deepening strategy similar to SAT-based planning [13]. If we want to reach a state satisfying $p \wedge \neg q$, we would include two constraints for the last situation:

$$\bot \leftarrow \mathit{not}\ p(n) \qquad\qquad \bot \leftarrow q(n)$$

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

[1] John McCarthy, the founder of logical knowledge representation and commonsense reasoning, showed in several occasions an explicit disapproval of modal logics. See for instance his position paper with the self-explanatory title "Modality, si! Modal logic, no!" [4].

[2] Generally speaking, a disjunction of the form $\varphi \vee \mathit{not}\ \varphi$ 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.

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.

In principle, one may think that this kind of problems dealing with infinite time are typically best suited for modal temporal logics, whose expressive power, computation methods (usually decidable) and associated complexity have been extensively well-studied. Unfortunately, as happens with SAT in the non-temporal case, temporal logics are not designed for Knowledge Representation (KR). For instance, the best known temporal logics are monotonic, so that the frame and ramification problems constantly manifest in their applications, even for very simple scenarios.

In this work, we make a general overview on *Temporal Equilibrium Logic* [14], to the best of our knowledge, the first non-monotonic approach that fully covers the syntax of some standard modal temporal logic, providing a logic programming semantics that properly extends *stable models* [15], the foundational basis of ASP. TEL shares the syntax of *Linear-time Temporal Logic* (LTL) [16,17] which is perhaps the simplest, most used and best known temporal logic in Theoretical Computer Science. The main difference of TEL with respect to LTL lies in its non-monotonic entailment relation (obtained by a models selection criterion) and in its semantic interpretation of implication and negation, closer to intuitionistic logic. These two properties are actually inherited from the fact that TEL is a temporal extension of *Equilibrium Logic* [18], a non-monotonic formalism that generalises stable models to the case of arbitrary propositional formulas. This semantic choice is a valuable feature because, on the one hand, it provides a powerful connection to a successful practical KR paradigm like ASP, and on the other hand, unlike the original definition of stable models, the semantics of Equilibrium Logic does not depend on syntactic transformations but, on the contrary, is just a simple minimisation criterion for an intermediate logic (the logic of *Here-and-There* [19]). This purely logical definition provides an easier and more homogeneous way to extend the formalism, using standard techniques from other hybrid logical approaches.

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

$$p \vee q \tag{5}$$

$$\Box(p \wedge \neg \bigcirc q \rightarrow \bigcirc p) \tag{6}$$

$$\Box(q \wedge \neg \bigcirc p \rightarrow \bigcirc q) \tag{7}$$

$$\Box(p \vee \neg p) \tag{8}$$

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 $\neg\Diamond(\neg p \wedge \neg q) \rightarrow \bot$ with '$\Diamond$' meaning "eventually." Similarly, to test whether p remains true after becoming true we would add the constraint $\Box(p \rightarrow \Box p) \rightarrow \bot$ and check that, indeed, no temporal stable model exists.

The rest of the paper is organised as follows. In Section 2 we recall the basic definitions of TEL and explain their effects on some simple examples. In Section 3 we summarize some fundamental properties whereas in Section 4 we explain some aspects related to computation. Finally, Section 5 concludes the paper and explains some open topics. For a more detailed survey, see [20].

2 Syntax and Semantics

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 $i \geq 0$.

The next step is defining a semantics for the temporal extension of the intermediate logic of Here-and-There, we will call *Temporal Here-and-There*[3] (THT). 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 (THT satisfaction). *Given an interpretation* $\mathbf{M} = \langle \mathbf{H}, \mathbf{T} \rangle$*, we recursively define when* $\mathbf{M}$ satisfies *a temporal formula* φ *at some state* $i \in \mathbb{N}$ *as:*

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

Proposition 1 (from [20]). *The following properties are satisfied:*

(i) $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \varphi$ *in THT iff* $\mathbf{T}, i \models \varphi$ *in LTL.*
(ii) $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \varphi$ *implies* $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \varphi$ *(that is,* $\mathbf{T}, i \models \varphi$*).*

In other words, (i) means that, when restricting to total interpretations, THT collapses to LTL, whereas (ii) means that the $\mathbf{T}$ component of a THT model is also an LTL-model.

[3] The axiomatisation of THT is currently under study [21].

Definition 2 (Temporal Equilibrium/Stable Model). *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*.* $\square$

By Proposition 1 (i) it is easy to see that 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 [22], a kind of 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 (5)-(8) 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.

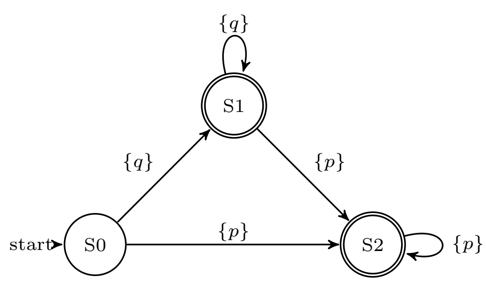

Fig. 1. Temporal stable models of theory (5)-(8).

Let us discuss next some simpler examples of the behaviour of this semantics. As a first example, consider the formula

$$\square(\neg p \rightarrow \bigcirc p) \tag{9}$$

Its intuitive meaning corresponds to the logic program consisting of rules of the form: $p(s(X)) \leftarrow \mathit{not}\ p(X)$ where time has been reified as an extra parameter $X = 0, s(0), s(s(0)), \ldots$. Notice that the interpretation of $\neg$ is that of default negation *not* in logic programming. In this way, (9) is saying that, at any situation, if there is no evidence on p, then p will become true in the next state. In the initial state, we have no evidence on p, so this will imply $\bigcirc p$. As a result $\bigcirc \bigcirc p$ will have no applicable rule and thus will be false by default, and so on. It is easy to see that the unique temporal stable model of (9) is captured by the formula $\neg p \wedge \square(\neg p \leftrightarrow \bigcirc p)$ and is shown in the automaton of Figure 2(a).

(a) TS-models of (9) (b) TS-models of $\Diamond p$

Fig. 2. A pair of Büchi automata showing TS-models.

As a second example, take the formula $\Diamond p$. This formula informally corresponds to an infinite disjunction $p \vee \bigcirc p \vee \bigcirc \bigcirc p \vee \dots$. Again, as happens in disjunctive logic programming, in TEL we have a truth minimality condition that will make true the formula with as little information as possible. As a result, it is easy to see that the temporal stable models of $\Diamond p$ are captured by the formula $\neg p \, \mathcal{U} (p \wedge \bigcirc \Box \neg p)$ whose models are those where p holds true at exactly one position – see automaton in Figure 2(b).

It is worth noting that an LTL satisfiable formula may have no temporal stable model. As a simple example (well-known from non-temporal ASP) the logic program rule $\neg p \rightarrow p$, whose only (classical) model is $\{p\}$, has no stable models. When dealing with logic programs, it is well-known that non-existence of stable models is always due to a kind of cyclic dependence on default negation like this. In the temporal case, however, non-existence of temporal stable models may also be due to a lack of a finite justification for satisfying the criterion of minimal knowledge. As an example, consider the formula $\alpha \stackrel{\text{def}}{=} \Box(\neg \bigcirc p \rightarrow p) \wedge \Box(\bigcirc p \rightarrow p)$. This formula has no temporal equilibrium models. To see why, note that α is LTL-equivalent (and THT-equivalent) to $\Box(\neg \bigcirc p \vee \bigcirc p \rightarrow p)$ that, in its turn, is LTL-equivalent to $\Box p$. Thus, the only LTL-model $\mathbf{T}$ of α has the form $T_i = \{p\}$ for any $i \geq 0$. However, it is easy to see that the interpretation $\langle \mathbf{H}, \mathbf{T} \rangle$ with $H_i = \emptyset$ for all $i \geq 0$ is also a THT model, whereas $\mathbf{H} < \mathbf{T}$.

Another example of TEL-unsatisfiable formula is $\Box \Diamond p$, typically used in LTL to assert that property p occurs infinitely often. This formula has no temporal stable models: all models must contain infinite occurrences of p and there is no way to establish a minimal $\mathbf{H}$ among them. Thus, formula $\Box \Diamond p$ is LTL satisfiable but it has no temporal stable model. This example does not mean a lack of expressiveness[4] of TEL: we can still check or force atoms to occur infinitely often by including formulas like $\Box \Diamond p$ in the antecedent of implications or in the scope of negation. As an example, take the formula:

$$\neg \Box \Diamond q \rightarrow \Diamond (q \, \mathcal{U} \, p) \qquad (10)$$

An informal reading of (10) is: if we cannot prove that q occurs infinitely often ($\neg \Box \Diamond q$) then make q until p ($q \, \mathcal{U} \, p$) at some arbitrary future point. As we minimise truth, we may then assume q false at all states, and then $\Diamond (q \, \mathcal{U} \, p)$

[4] In fact, Theorem 1 in the next section shows that LTL can be encoded into TEL by adding a simple axiom schema.

collapses to $\Diamond(\bot\ \mathcal{U}\ p) = \Diamond(\Diamond p) = \Diamond p$. As a result, its TS-models also correspond to the Büchi automaton depicted in Figure 2(b) we obtained for $\Diamond p$.

3 Fundamental properties

We first begin providing some translation results relating TEL and LTL.

Proposition 2 (from [23]). *The LTL models of a formula φ for signature At coincide with (the THT) and the TEL models of the theory φ plus an axiom $\Box(p \vee \neg p)$ for each atom p in the signature At.* □

The translation from THT to LTL is not so straightforward. It requires adding an auxiliary atom p' by each atom p in the signature, so that the former captures the truth at component $\mathbf{H}$ in a THT model $\langle \mathbf{H}, \mathbf{T} \rangle$ while the latter represents truth at $\mathbf{T}$. Given a propositional signature At, let us denote $At^* = At \cup \{p' \mid p \in At\}$. For any temporal formula φ we define its translation φ^* as follows:

1. $\bot^* \stackrel{\text{def}}{=} \bot$
2. $p^* \stackrel{\text{def}}{=} p'$ for any $p \in \Sigma$
3. $(\otimes\varphi)^* \stackrel{\text{def}}{=} \otimes\varphi^*$, for any unary operator $\otimes \in \{\Box, \Diamond, \bigcirc\}$
4. $(\varphi \oplus \psi)^* \stackrel{\text{def}}{=} \varphi^* \oplus \psi^*$ for any binary operator $\oplus \in \{\wedge, \vee, \mathcal{U}, \mathcal{R}\}$
5. $(\varphi \to \psi)^* \stackrel{\text{def}}{=} (\varphi \to \psi) \wedge (\varphi^* \to \psi^*)$

We associate to any THT interpretation $\mathbf{M} = \langle \mathbf{H}, \mathbf{T} \rangle$ the LTL interpretation $\mathbf{M}^t = \overline{I}$ in LTL defined as the sequence of sets of atoms $I_i = \{p' \mid p \in H_i\} \cup T_i$, for any $i \geq 0$.

Theorem 1 (from [23]). *Let φ' be the formula $\varphi^* \wedge \bigwedge_{p \in At} \Box(p' \to p)$. Then the set of LTL models for the formula φ' corresponds to the set of THT models for the temporal formula φ.* □

Theories like (5)-(8) have a strong resemblance to logic programs. For instance, a rule preceded by $\Box$ like (9) can be seen as an infinite set of rules of the form $\neg\bigcirc^i p \to \bigcirc^{i+1} p$ where we could understand expressions like '$\bigcirc^i p$' as an infinite propositional signature. In [24] it was recently proved that, in fact, we can use this understanding of modal operators as formulas in Infinitary Equilibrium Logic (see [25] for further detail) in the general case.

Definition 3. *The translation of φ into infinitary HT (HT^∞) up to level $k \geq 0$, written $\langle\varphi\rangle_k$, is recursively defined as follows:*

$\langle\bot\rangle_k \stackrel{\text{def}}{=} \emptyset^\vee$

$\langle p\rangle_k \stackrel{\text{def}}{=} \bigcirc^k p$, *with* $p \in At$.

$\langle\bigcirc\varphi\rangle_k \stackrel{\text{def}}{=} \langle\varphi\rangle_{k+1}$

$\langle\varphi \wedge \psi\rangle_k \stackrel{\text{def}}{=} \{\langle\varphi\rangle_k, \langle\psi\rangle_k\}^\wedge$

$\langle\varphi \vee \psi\rangle_k \stackrel{\text{def}}{=} \{\langle\varphi\rangle_k, \langle\psi\rangle_k\}^\vee$

$\langle\varphi \to \psi\rangle_k \stackrel{\text{def}}{=} \langle\varphi\rangle_k \to \langle\psi\rangle_k$

$\langle\varphi\ \mathcal{U}\ \psi\rangle_k \stackrel{\text{def}}{=} \{\{\langle\psi\rangle_i, \langle\varphi\rangle_j \mid k \leq j < i\}^\wedge \mid k \leq i\}^\vee$

$\langle\varphi\ \mathcal{R}\ \psi\rangle_k \stackrel{\text{def}}{=} \{\{\langle\psi\rangle_i, \langle\varphi\rangle_j \mid k \leq j < i\}^\vee \mid k \leq i\}^\wedge$

It is easy to see that the derived operators $\Box$ and $\Diamond$ are then translated as follows: $\langle \Diamond\varphi \rangle_k = \{\langle \varphi \rangle_i \mid k \leq i\}^\vee$ and $\langle \Box\varphi \rangle_k = \{\langle \varphi \rangle_i \mid k \leq i\}^\wedge$. For instance, the translations for our examples $\langle \Diamond p \rangle_0$ and $\langle (10) \rangle_0$ respectively correspond to:

$$\{\neg \bigcirc^i p \rightarrow \bigcirc^{i+1} p \mid i \geq 0\}^\wedge$$
$$\{\{\{\bigcirc^k q \mid j \leq k\}^\vee \mid i \leq j\}^\wedge \rightarrow \{\{\bigcirc^k p, \bigcirc^h q \mid j \leq h < k\}^\wedge \mid i \leq j \leq k\}^\vee \mid i \geq 0\}^\wedge$$

Theorem 2 (from [24]). *Let φ be a temporal formula formula, $\mathbf{M} = \langle \mathbf{H}, \mathbf{T} \rangle$ a THT interpretation and $M^\infty = \langle H^\infty, T^\infty \rangle$ its corresponding HT interpretation where $\bigcirc^i p$ are considered as propositional atoms. For all $i \in \mathbb{N}$, it holds that:*

(i) $\mathbf{M}, i \models \varphi$ *if and only if* $M^\infty \models \langle \varphi \rangle_i$.
(ii) $\mathbf{M}$ *is a temporal equilibrium model of φ if and only if M^∞ is an (infinitary) equilibrium model of* $\langle \varphi \rangle_0$. □

In [24] it was also proved that Kamp's translation from LTL to First Order Logic is sound for translating TEL into Quantified Equilibrium Logic [26] too. This means that there always exists a way of resorting to first-order ASP and reifying time as an argument, as we did before with $p(i)$ or $p(i + 1)$, so that modal operators are replaced by standard quantifiers.

Definition 4 (Kamp's translation). *Kamp's translation for a temporal formula φ and a timepoint $t \in \mathbb{N}$, denoted by $[\varphi]_t$, is recursively defined as follows:*

$$[\bot]_t \stackrel{\text{def}}{=} \bot$$
$$[p]_t \stackrel{\text{def}}{=} p(t), \text{ with } p \in At.$$
$$[\neg\alpha]_t \stackrel{\text{def}}{=} \neg[\alpha]_t$$
$$[\alpha \wedge \beta]_t \stackrel{\text{def}}{=} [\alpha]_t \wedge [\beta]_t$$
$$[\alpha \vee \beta]_t \stackrel{\text{def}}{=} [\alpha]_t \vee [\beta]_t$$
$$[\alpha \rightarrow \beta]_t \stackrel{\text{def}}{=} [\alpha]_t \rightarrow [\beta]_t$$
$$[\bigcirc\alpha]_t \stackrel{\text{def}}{=} [\alpha]_{t+1}$$
$$[\alpha\ \mathcal{U}\ \beta]_t \stackrel{\text{def}}{=} \exists x \geq t.\ ([\beta]_x \wedge \forall y \in [t, x).\ [\alpha]_y)$$
$$[\alpha\ \mathcal{R}\ \beta]_t \stackrel{\text{def}}{=} \forall x \geq t.\ ([\beta]_x \vee \exists y \in [t, x).\ [\alpha]_y)$$

where $[\alpha]_{t+1}$ *is an abbreviation of* $\exists y \geq t.\ \neg\exists z \in [t, y).\ (t < z \wedge [\alpha]_y)$. □

Note how, per each atom $p \in At$ in the temporal formula φ, we get a monadic predicate $p(x)$ in the translation. The effect of this translation on the derived operators $\Diamond$ and $\Box$ yields the quite natural expressions $[\Box\alpha]_t \equiv \forall x \geq t.\ [\alpha]_t$ and $[\Diamond\alpha]_t \equiv \exists x \geq t.\ [\alpha]_t$. For instance, the translations of our running examples (9) and (10) for $t = 0$ respectively correspond to:

$$\forall x \geq 0.\ (\neg p(x) \rightarrow p(x + 1)) \tag{11}$$
$$\forall x \geq 0.\ \Big(\forall y \geq x.\ \exists z \geq y.\ q(z) \rightarrow \exists y \geq x.\ \exists z \geq y.\ \big(p(z) \wedge \forall t \geq y.\ zq(t)\big)\Big) \tag{12}$$

Theorem 3 (from [24]). *Let φ be a THT formula built on a set of atoms At, $\mathbf{M} = \langle \mathbf{H}, \mathbf{T} \rangle$ a THT-interpretation on At and $\mathcal{M} = \langle \mathcal{H}, \mathcal{T} \rangle$ its corresponding Quantified HT-interpretation. It holds that $\mathbf{M}, i \models \varphi$ in THT iff $\mathcal{M} \models [\varphi]_i$ in Quantified Here-and-There. Moreover, $\mathbf{T}$ is a TS-model of φ iff $\mathcal{T}$ is a stable model of $[\varphi]_0$ in Quantified Equilibrium Logic.* □

Another group of properties is related to comparison among temporal theories and subclasses of theories. For instance, 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). [27] 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 $\Gamma_1 \equiv_{HT} \Gamma_2$. 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 [28] it was shown that something similar happens in the temporal case, namely:

Theorem 4 (from [28]). *Two temporal formulas α and β are strongly equivalent in TEL iff they are THT-equivalent.* □

Another interesting result related to equivalence is the existence of normal forms for THT and TEL. In the case of Equilibrium Logic, it has been already proved [29] that any arbitrary propositional theory is strongly equivalent to a logic program (allowing disjunction and negation in the head). Similarly, in the case of (monotonic) LTL, an implicational clause-like normal form introduced in [30] was used for designing a temporal resolution method.

Following [31], TEL can be similarly reduced (under strong equivalence) to a normal form, called *temporal logic programs* (TLP), consisting of a set of implications (embraced by a necessity operator) quite close to logic program rules. The obtained normal form considerably reduces the possible uses of modal operators and, as we will see later, has became useful for a practical computation of TEL models. The definitions are as follows. Given a signature At, we define a *temporal literal* as any expression in the set $\{p, \bigcirc p, \neg p, \neg \bigcirc p \mid p \in At\}$.

Definition 5 (Temporal rule). *A* temporal rule *is either:*

1. *an* initial rule *of the form $B_1 \wedge \cdots \wedge B_n \rightarrow C_1 \vee \cdots \vee C_m$ where all the B_i and C_j are temporal literals, $n \geq 0$ and $m \geq 0$.*
2. *a* dynamic rule *of the form $\Box r$, where r is an initial rule.*
3. *a* fulfillment rule *like $\Box(\Box p \rightarrow q)$ or like $\Box(p \rightarrow \Diamond q)$ with p, q atoms.* □

In the three cases, we respectively call rule *body* and rule *head* to the antecedent and consequent of the (unique) rule implication. In initial (resp. dynamic) rules, we may have an empty head $m = 0$ corresponding to $\bot$ – if so, we talk about an *initial* (resp. *dynamic*) *constraint.* A *temporal logic program*[5] (TLP for short) is

[5] In fact, as shown in [31], this normal form can be even more restrictive: initial rules can be replaced by atoms, and we can avoid the use of literals of the form $\neg\bigcirc p$.

a finite set of temporal rules. The reduction into TLP normal form introduces an auxiliary atom per each subformula in the original theory and applies the inductive definitions of temporal operators used for LTL in [30]. We will not enter into further details (see [31]) but the obtained reduction into TLP is modular, polynomial and strongly faithful (that is, it preserves strong equivalence, if auxiliary atoms are ignored).

4 Computation

Computation of TS-models is a complex task. THT-satisfiability has been classified [23] as PSPACE-complete, that is, the same complexity as LTL-satisfiability, whereas TEL-satisfiability rises to EXPSPACE-completeness, as recently proved in [32]. In this way, we face a similar situation as in the non-temporal case where HT-satisfiability is NP-complete like SAT, whereas existence of equilibrium model (for arbitrary theories) is Σ_2^P-complete (like disjunctive ASP).

There exists a pair of tools, `STeLP` [33] and `ABSTEM` [28], that allow computing temporal stable models (represented as Büchi automata). These tools can be used to check verification properties that are usual in LTL, like the typical safety, liveness and fairness conditions, but in the context of temporal ASP. Moreover, they can also be applied for planning problems that involve an indeterminate or even infinite number of steps, such as the non-existence of a plan.

The first tool, `STeLP`, accepts a strict subset of the TLP normal form called *splittable* temporal formulas (STF) which will be of one of the following types:

$$B \wedge N \rightarrow H \tag{13}$$

$$B \wedge \bigcirc B' \wedge N \wedge \bigcirc N' \rightarrow \bigcirc H' \tag{14}$$

$$\Box(B \wedge \bigcirc B' \wedge N \wedge \bigcirc N' \rightarrow \bigcirc H') \tag{15}$$

where B and B' are conjunctions of atomic formulas, N and N' are conjunctions of $\neg p$, being p an atomic formula and H and H' are disjunctions of atomic formulas.

The name *splittable* refers to the fact that these programs can be splitted using [34] thanks to the property that rule heads never refer to a time point previous to those referred in the body. As we can see above, the main property of a splittable temporal rule is that, informally speaking, *past never depends on the future*, that is, we never get references to $\bigcirc$ in the rule bodies unless all atoms in the head are also in the scope of $\bigcirc$. As shown in [35], when the input temporal program is splittable, it is possible to extend the technique of *loop formulas* [36] to temporal theories so that it is always possible to capture the TS-models of a theory Γ as the LTL-models of another theory Γ' obtained from Γ together with its loop formulas. Although splittable theories do not cover the full expressiveness of TEL, most action domains represented in ASP are indeed splittable. To cover an ASP-like syntax, `STeLP` further allows the use of variables: a preliminary grounding method was presented in [37], proving its correctness.

The tool ABSTEM, on the contrary, accepts any arbitrary temporal theory as an input, although it does not accept variables. It relies on an automata-based transformation described in [23] and it not only allows computing the TS-models of a temporal theory, but also accepts pairs of theories to decide different types of equivalence: LTL-equivalence, TEL-equivalence (i.e. coincidence in the set of TS-models) and strong equivalence (i.e., THT-equivalence). Moreover, when strong equivalence fails, ABSTEM obtains a context, that is, an additional formula that added to the compared theories makes them behave differently.

5 Conclusions and future work

In this survey we have summarised the basic results on Temporal Equilibrium Logic obtained so far, showing that it can be used as a powerful tool for combining temporal reasoning tasks with Answer Set Programming. Still, there are many open topics that deserve to be studied. For instance, in the theoretical setting, we still miss a complete axiomatisation of THT. Another open question is that, although we know that Kamp's translation from LTL into First Order Logic also works for translating TEL into Quantified Equilibrium Logic, we ignore whether the other direction of Kamp's theorem also holds in this case. Namely, we ignore whether any theory in Monadic Quantified Equilibrium Logic for a linear order relation $<$ can be represented in TEL. A possibly related question is whether the set of TS-models of a temporal theory can be captured as the set of LTL-models of another theory. This holds in the case of splittable temporal logic programs, but is open in the general case[6].

An interesting research line is the extension of TEL with past operators, since they seem more natural for rule bodies that describe the transitions of a dynamic system. Besides, following similar steps as those done in TEL, other hybrid approaches can be explored. For instance, [39] has considered the combination of Dynamic LTL with Equilibrium Logic. Similarly, other temporal approaches can be treated in an analogous way, such as CTL, CTL*, Dynamic Logic or μ-calculus. Other open topics are related to potential applications including translation of different action languages, policy languages with preferences [40] or planning with (temporal) control rules.

Acknowledgements This research is part of a long term project developed during the last eight years in the KR group from the University of Corunna and, especially, in close cooperation with Felicidad Aguado, Martín Diéguez, Gilberto Pérez and Concepción Vidal together with the regular collaborators David Pearce and Luis Fariñas. I am also especially thankful to Stèphane Demri, Philippe Balbiani, Andreas Herzig, Laura Bozzelli, Manuel Ojeda, Agustín Valverde, Stefania Costantini, Michael Fisher, Mirosław Truszczyński, Vladimir Lifschitz and Torsten Schaub for their useful discussions and collaboration at different moments on specific topics of this work.

[6] An incorrect proof was published in [38], but we still conjecture that the result might be positive.

References

1. McCarthy, J., Hayes, P.J.: Some philosophical problems from the standpoint of artificial intelligence. In: Machine Intelligence, Edinburgh University Press (1969) 463–502
2. Kowalski, R., Sergot, M.: A logic-based calculus of events. New Generation Computing **4** (1986) 67–95
3. McCarthy, J.: Circumscription: A Form of Non-Monotonic Reasoning. Artificial Intelligence **13** (1980) 27–39
4. Mccarthy, J.: Modality, si! modal logic, no! Studia Logica **59**(1) (1997) 29–32
5. Castilho, M.A., Gasquet, O., Herzig, A.: Formalizing action and change in modal logic I: the frame problem. Journal of Logic and Computation **9**(5) (1999) 701–735
6. Giordano, L., Martelli, A., Schwind, C.: Ramification and causality in a modal action logic. Journal of Logic and Computation **10**(5) (2000) 625–662
7. Baral, C., Zhao, J.: Nonmonotonic temporal logics for goal specification. In: Proc. of the 20th Intl. Joint Conf. on Artificial Intelligence (IJCAI'07). (2007) 236–242
8. Gelfond, M., Lifschitz, V.: Representing action and change by logic programs. The Journal of Logic Programming **17** (1993) 301–321
9. Gelfond, M., Lifschitz, V.: Action languages. Linköping Electronic Articles in Computer and Information Science **3**(16) (1998) (`http://www.ep.liu.se/ea/cis/1998/016`).
10. Niemelä, I.: Logic Programs with Stable Model Semantics as a Constraint Programming Paradigm. Annals of Mathematics and Artificial Intelligence **25**(3-4) (1999) 241–273
11. Marek, V., Truszczyński, M. In: Stable models and an alternative logic programming paradigm. Springer-Verlag (1999) 169–181
12. Gebser, M., Sabuncu, O., Schaub, T.: An incremental answer set programming based system for finite model computation. AI Commun. **24**(2) (2011) 195–212
13. Kautz, H.A., Selman, B.: Planning as satisfiability. In: Proc. of the European Conf. on Artificial Intelligence (ECAI'92). (1992) 359–363
14. Cabalar, P., Perez, G.: Temporal Equilibrium Logic: A First Approach. In: Proceedings of the 11^{th} International Conference on Computer Aided Systems Theory (EUROCAST'07). (2007) 241248
15. Gelfond, M., Lifschitz, V.: The Stable Model Semantics For Logic Programming. In: Proc. of the 5^{th} International Conference on Logic Programming (ICLP'88), Seattle, Washington (1988) 10701080
16. Prior, A.: Past, Present and Future. Oxford University Press (1967)
17. Kamp, H.: Tense Logic and the Theory of Linear Order. PhD thesis, UCLA (1968)
18. Pearce, D.: A New Logical Characterisation of Stable Models and Answer Sets. In: Proc. of Non-Monotonic Extensions of Logic Programming (NMELP'96), Bad Honnef, Germany (1996) 57–70
19. Heyting, A.: Die formalen Regeln der intuitionistischen Logik. Sitzungsberichte der Preussischen Akademie der Wissenschaften. Physikalisch-mathematische Klasse (1930)
20. Aguado, F., Cabalar, P., Diéguez, M., Pérez, G., Vidal, C.: Temporal equilibrium logic: a survey. Journal of Applied Non-Classical Logics **23**(1-2) (2013) 2–24
21. Balbiani, P., Diéguez, M.: An axiomatisation of the logic of temporal here-and-there (2015) (unpublished draft).
22. Büchi, J.R.: On a Decision Method in Restricted Second-Order Arithmetic. In: International Congress on Logic, Methodology, and Philosophy of Science. (1962) 1–11

23. Cabalar, P., Demri, S.: Automata-based computation of temporal equilibrium models. In: 21st International Symposium on Logic-Based Program Synthesis and Transformation (LOPSTR'11). (2011)
24. Cabalar, P., Diéguez, M., Vidal, C.: An infinitary encoding of temporal equilibrium logic. In: Proc. of the 31st Intl. Conf. on Logic Programming (ICLP'15). (2015)
25. Harrison, A., Lifschitz, V., Pearce, D., Valverde, A.: Infinitary equilibrium logic. In: Working Notes of Workshop on Answer Set Programming and Other Computing Paradigms (ASPOCP'14). (2014)
26. Pearce, D., Valverde, A.: Quantified equilibrium logic and foundations for answer set programs. In: Proc. of the 24th Intl. Conf. on Logic Programming, ICLP 2008, (Udine, Italy, December 9-13). Volume 5366 of Lecture Notes in Computer Science. Springer (2008) 546–560
27. Lifschitz, V., Pearce, D., Valverde, A.: Strongly equivalent logic programs. Computational Logic **2**(4) (2001) 526–541
28. Cabalar, P., Diéguez, M.: Strong equivalence of non-monotonic temporal theories. In: Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14), Vienna, Austria (2014)
29. Cabalar, P., Ferraris, P.: Propositional theories are strongly equivalent to logic programs. Theory and Practice of Logic Programming **7**(6) (2007) 745–759
30. Fisher, M.: A resolution method for temporal logic. In: Proceedings of the 12th international joint conference on Artificial Intelligence (IJCAI'91), Morgan Kaufmann Publishers Inc. (1991) 99–104
31. Cabalar, P.: A normal form for linear temporal equilibrium logic. In: Proc. of the 12th European Conference on Logics in Artificial Intelligence (JELIA'10). Lecture Notes in Computer Science. Volume 2258. Springer-Verlag (2010) 64–76
32. Bozzelli, L., Pearce, D.: On the complexity of temporal equilibrium logic. In: Proceedings of the 30th Annual ACM/IEEE Symposium of Logic in Computer Science (LICS'15), Kyoto, Japan (2015) (to appear).
33. Cabalar, P., Diéguez, M.: STELP - a tool for temporal answer set programming. In: LPNMR'11. Volume 6645 of Lecture Notes in Computer Science. (2011) 370–375
34. Lifschitz, V., Turner, H.: Splitting a logic program. In: Proceedings of the 11th International Conference on Logic programming (ICLP'94). (1994) 23–37
35. Aguado, F., Cabalar, P., Pérez, G., Vidal, C.: Loop formulas for splitable temporal logic programs. In Delgrande, J.P., Faber, W., eds.: LPNMR'11. Volume 6645 of Lecture Notes in Computer Science., Springer (2011) 80–92
36. Ferraris, P., Lee, J., Lifschitz, V.: A generalization of the Lin-Zhao theorem. Annals of Mathematics and Artificial Intelligence **47** (2006) 79–101
37. Aguado, F., Cabalar, P., Diéguez, M., Pérez, G., Vidal, C.: Paving the way for temporal grounding. In: Proc. of the 28th International Conference on Logic Programming (ICLP'12). (2012)
38. Cabalar, P., Diéguez, M.: Temporal stable models are LTL-representable. In: Proc. of the 7th Intl. Workshop on Answer Set Programming and Other Computing Paradigms (ASPOCP'14). (2014)
39. Aguado, F., Pérez, G., Vidal, C.: Integrating Temporal Extensions of Answer Set Programming. In: Proc. of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'13), Corunna, Spain (2013) 23–35
40. Bertino, E., Mileo, A., Provetti, A.: PDL with preferences. In: 6th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2005). (2005) 213–222