Strong Equivalence of Non-monotonic Temporal Theories[*]

Pedro Cabalar and **Martín Diéguez**
Dept. of Computer Science
University of Corunna, Spain
{cabalar,martin.dieguez}@udc.es

Abstract

In this paper we solve the following open problem: we prove that equivalence in the logic of *Temporal Here-and-There* (THT) is not only a sufficient, but also a *necessary* condition for strong equivalence of two *Temporal Equilibrium Logic* (TEL) theories. This result 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

With the consolidation of Answer Set Programming (ASP) (Brewka, Eiter, and Truszczyński 2011) as a successful paradigm for practical knowledge representation, many examples and benchmarks formalising dynamic scenarios became available. ASP usually treats time as an integer index restricting all reasoning tasks to finite narratives. 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 either p or q holds initially. A typical ASP representation could be:

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

where (1) is the initial state, (2) and (3) are the inertia rules and (4) is a choice rule for p at any situation. Predicate sit would have a finite domain $0 \ldots n$ for some constant $n \geq 0$. Planning for a goal like $p \wedge \neg q$ implies including two constraints for the last situation, $\bot \leftarrow not\, p(n)$ and $\bot \leftarrow q(n)$, and go increasing n until a solution is found. 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 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, (Aguado et al. 2013) 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) (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. For instance, the ASP program (1)-(4) would be represented in TEL as:

$$p \vee q \tag{5}$$
$$\Box(p \wedge \neg \bigcirc q \to \bigcirc p) \tag{6}$$
$$\Box(q \wedge \neg \bigcirc p \to \bigcirc q) \tag{7}$$
$$\Box(p \vee \neg p) \tag{8}$$

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:

$$\neg \Diamond(\neg p \wedge \neg q) \to \bot \tag{9}$$

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

$$\Box(p \to \Box p) \to \bot \tag{10}$$

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) – but a few important questions about TEL remained unsolved. One of them has to do with the property of *strong equivalence* in TEL. In any Non-Monotonic Reasoning approach, 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 (stable models in ASP). (Lifschitz, Pearce, and Valverde 2001) proved that checking equivalence in the logic of HT 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). This result for propositional HT was 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.

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[1], 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.

The rest of the paper is organised as follows. In the next section we recall the basic definitions of TEL. After that, we overview the automata-based techniques from (Cabalar and Demri 2011). The next section contains the main theorem. This is followed by an explanation of the implementation together with a practical example. Finally, we include some conclusions.

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 $\Box\varphi \overset{\text{def}}{=} \bot\, \mathcal{R}\, \varphi$ and $\Diamond\varphi \overset{\text{def}}{=} \top\, \mathcal{U}\, \varphi$. Other

usual propositional operators are defined as follows: $\neg\varphi \overset{\text{def}}{=} \varphi \rightarrow \bot$, $\top \overset{\text{def}}{=} \neg\bot$ and $\varphi \leftrightarrow \psi \overset{\text{def}}{=} (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.

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.*

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 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 (5)-(8) which coincide with sequences of states of the forms $\{q\}^*\{p\}^\omega$ or $\{q\}^\omega$.

Computing Temporal Stable Models of Arbitrary Theories

Based on the techniques introduced in (Cabalar and Demri 2011), the tool ABSTEM constitutes the first implementation capable of computing TS-models for arbitrary temporal formulas, without syntactic restrictions. The 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
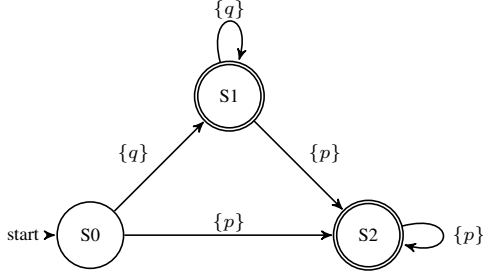
Figure 1: Temporal stable models of theory (5)-(8).

$\langle \mathbf{T}, \mathbf{T} \rangle$ of φ. By Proposition 1 this simply amounts to compute the LTL models $\mathbf{T}$ of φ using an automata construction method for LTL. 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 complementary 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). It uses an extended signature $\{p' \mid p \in At\}$ so that p' represents the truth of p in $\mathbf{H}$ while p is used for $\mathbf{T}$. The translation of φ, written φ^*, is recursively defined as follows:

- $(p)^* \stackrel{\text{def}}{=} p'$, for any atom $p \in At$

- $(\alpha \rightarrow \beta)^* \stackrel{\text{def}}{=} (\alpha \rightarrow \beta) \wedge (\alpha^* \rightarrow \beta^*)$

being homomorphic for the rest of logical connectives. To impose the restriction $\mathbf{H} < \mathbf{T}$ we further include the axiom:

$$\left(\bigwedge_{p \in At} \square(p' \rightarrow p) \right) \wedge \left(\bigvee_{p \in At} \Diamond(\neg p' \wedge p) \right) \quad \textbf{(Ax1)}$$

Automaton $A_{\varphi''}$ is built from formula $\varphi'' \stackrel{\text{def}}{=} \varphi^* \wedge (\textbf{Ax1})$. The operation $h(A_{\varphi''})$ returns a new automaton that results from removing the atoms p' from transitions in $A_{\varphi''}$. This 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.

Temporal Strong Equivalence

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

Theorem 1 (Sufficient condition (Aguado et al. 2008))
If two temporal formulas α and β are THT-equivalent then they are strongly equivalent in TEL. ∎

To prove the other direction, namely, that THT-equivalence is also a necessary condition for strong equivalence, we begin defining γ_0 as the conjunction of all formulas $\square(p \vee \neg p)$ for all atoms $p \in At$.

Proposition 2 *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 $\gamma = (\beta \rightarrow \gamma_0)$. Then, the following conditions are equivalent:*

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

(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 We will prove that 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 (since γ_0 is an LTL tautology). But by Corollary 1, LTL-models of these theories are exactly their TS-models, and so, they 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 $(\alpha \rightarrow \beta)$ or $(\beta \rightarrow \alpha)$, and given LTL-equivalence of α and β, the countermodel is non-total, $\mathbf{H} < \mathbf{T}$. Without loss of generality, assume $\langle \mathbf{H}, \mathbf{T} \rangle \not\models \alpha \rightarrow \beta$. By Lemma 1, taking the formula $\gamma = (\beta \rightarrow \gamma_0)$, we get that $\mathbf{T}$ is TS-model of $\beta \wedge \gamma$ but not TS-model of $\alpha \wedge \gamma$. ∎

Algorithm 1 StrongEquivalenceTest(α, β)

Require: Two propositional temporal formulas α, β.
Ensure: If α and β are THT-equivalent, it returns **true**. Otherwise, it returns a triple $\langle \gamma, A_1, A_2 \rangle$.

$A_1 := \text{ltl\_to\_Büchi}(\alpha \wedge \neg\beta)$
$A_2 := \text{ltl\_to\_Büchi}(\beta \wedge \neg\alpha)$
if $A_1 \neq \emptyset$ **or** $A_2 \neq \emptyset$ **then**
    return $\langle \gamma_0, A_1, A_2 \rangle$
end if
$A = \text{ltl\_to\_Büchi}(\neg(\alpha \rightarrow \beta)^* \wedge (\textbf{Ax1}))$
if $A \neq \emptyset$ **then**
    $A_2 := h(A)$
    return $\langle (\beta \rightarrow \gamma_0), \emptyset, A_2 \rangle$
end if
$A = \text{ltl\_to\_Büchi}(\neg(\beta \rightarrow \alpha)^* \wedge (\textbf{Ax1}))$
if $A \neq \emptyset$ **then**
    $A_1 := h(A)$
    return $\langle (\alpha \rightarrow \gamma_0), A_1, \emptyset \rangle$
end if
return true

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[2]. 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 an auxiliary routine ltl_to_Büchi(φ) to obtain a Büchi automaton from an LTL-formula φ.

As an example of use, let β_1 be our "switch" domain (5)-(8) plus the rule $\Box(\neg p \to q)$ 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 (6),(7) destroying somehow their effect. Using ABSTEM to check the TS-models of β_1 we obtain the automaton in Figure 2(a) which corresponds to arbitrary sequences formed with states $\{p\}$ and $\{q\}$. 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. So, we may conjecture that β_1 is equivalent to $\alpha_1 = \Box(p \vee q)$. Using ABSTEM we can check that, in fact, α_1 and β_1 have the same TS-models and, furthermore, they are also LTL-equivalent. However, α_1 and β_1 are not THT-equivalent and so, they are not strongly equivalent. The answer displayed this time by ABSTEM is negative and shows the context formula $\gamma_1 = \beta_1 \to \gamma_0$ plus a file containing the automaton in Figure 2(b). This automaton captures all the TS-models of $\beta_1 \wedge \gamma_1$ that are not TS-models of $\alpha_1 \wedge \gamma_1$.
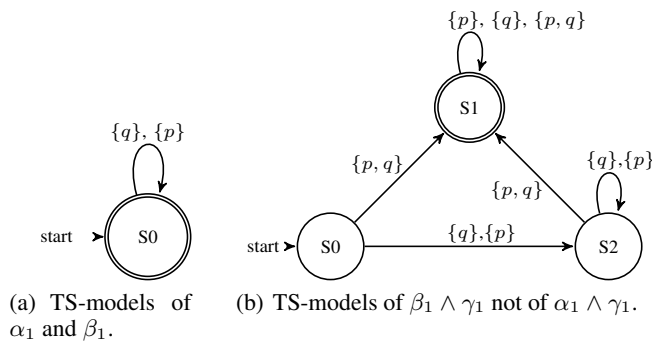

(a) TS-models of α_1 and β_1.

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

Figure 2: Temporal stable models related to α_1 and β_1.

Conclusions

In this paper we have adapted (Lifschitz, Pearce, and Valverde 2007) to the temporal case, to prove that equivalence in the logic of Temporal Here-and-There (THT), is not only a sufficient but also a *necessary* condition for strong equivalence in Temporal Equilibrium Logic (TEL). Using this proof, we have implemented a system, ABSTEM, for analysing TEL arbitrary theories in different ways. First, ABSTEM constitutes the first tool for computing temporal equilibrium models of any arbitrary temporal theory. Second, it also allows checking three types of equivalence between two arbitrary theories: LTL, weak and strong equivalence. When equivalence is not satisfied, ABSTEM shows

counterexamples in the form of a Büchi automaton. Regarding efficiency and scalability, this prototype works satisfactorily for small theories like the ones presented in the paper. It must be noted, however, that checking THT-equivalence is a PSPACE-complete problem.

References

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, 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.

Brewka, G.; Eiter, T.; and Truszczyński, M. 2011. Answer set programming at a glance. *Commun. ACM* 54(12):92–103.

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, P., and Demri, S. 2011. Automata-based computation of temporal equilibrium models. In Vidal, G., ed., *LOPSTR'11*, Lecture Notes in Computer Science. Odense, Denmark: Springer.

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

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

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

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.

Pearce, D.; Tompits, H.; and Woltran, S. 2001. Encodings for equilibrium logic and logic programs with nested expressions. In *EPIA 2001*, volume 2258 of *Lecture Notes in Computer Science*, 306–320. Springer.

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

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

---

[2]For further details, see the extended version of this document at `http://kr.irlab.org/sites/default/files/papers/kr2014-ex.pdf`.