Temporal Stable Models are LTL-representable

Pedro Cabalar and Martín Diéguez

Department of Computer Science
University of Corunna (Spain)
{cabalar, martin.dieguez}@udc.es

Abstract. Many scenarios in Answer Set Programming (ASP) deal with dynamic systems over (potentially infinite) linear time. Temporal Equilibrium Logic (TEL) is a formalism that allows defining the idea of *temporal stable model* not only for dynamic domains in ASP, but aso for any arbitrary theory in the syntax of Linear-Time Temporal Logic (LTL). In the past, several tools for computing temporal stable models have been built using well-established LTL and automata techniques. These tools displayed the set of temporal stable models of a given theory as a Büchi-automaton and, in many cases, it was also possible to capture such a set by the LTL-models of a given temporal formula. The fundamental theoretical question of whether this was a general property or not remained open, since it is well-known that, in general, Büchi-automata are more expressive than LTL. In this paper we show that, indeed, the set of temporal stable models of any arbitrary temporal theory can be succinctly captured as the LTL models of a given temporal formula.

1 Introduction

Temporal Equilibrium Logic [1] is a hybrid formalism mixing *Equilibrium Logic* [2,3], the best-known logical characterisation of Answer Set Programming (ASP), with *Linear-Time Temporal Logic* [4,5] (LTL), one of the simplest and most extensively studied modal temporal logics. TEL is suitable for studying temporal properties of ASP specifications of dynamic scenarios such as, for instance, checking the non-existence of a plan, searching for plans that satisfy temporal constraints, checking liveness and safety conditions or checking strong equivalence of theories with temporal constraints.

To illustrate how a TEL encoding of an ASP program may look like, consider a program containing the single rule:

$$p(I+1) \leftarrow \textit{not}\ p(I)$$

where I is an integer time variable ranging in some (usually finite) domain $0, 1, \ldots, n$. A program like this would make $p(0)$ false, since no fact is given for the initial state, then $p(1)$ true by applying the rule with $I = 1$, then $p(2)$ false because no rule can be applied for $I = 2$ and so on. The translation of this rule into TEL would look like:

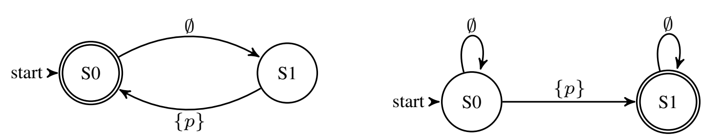
$$\Box(\neg p \rightarrow \bigcirc p) \tag{1}$$

where, as usual in LTL, '$\Box$' means "always" and '$\bigcirc$' stands for "in the next state," whereas time is understood as an infinite linear sequence of states (propositional interpretations). TEL allows extending the concept of stable model for any temporal theory in the syntax of LTL. A *temporal stable model* (for short, *TS-model*) will have the form of a temporal interpretation, that is, an infinite sequence of sets of atoms or states. In particular, (1) has a unique TS-model formed by the infinite alternating sequence of states $\emptyset, \{p\}, \emptyset, \{p\}, \ldots$ A pair of tools for computing the TS-models of a temporal theory have been already built. The first of them, `STeLP` [6], accepts a syntactic (strict) subset of TEL, the so-called *splittable* temporal logic programs. A theory is a "splittable program" iff, informally speaking, it consists of temporal rules that do not introduce a dependence from future to the past (see [7] for further details). This syntactic subset, which covers most dynamic scenarios in the ASP literature, has an important advantage: their TS-models are LTL-*representable*.

Definition 1. *We say that a set S of temporal interpretations is LTL-representable iff S is the set of LTL-models of some temporal formula φ.* ∎

TS-models of any splittable temporal logic program Π are LTL-representable. This is because, when Π is splittable, it is always possible to apply well-known ASP techniques like splitting and loop formulas to get a formula φ whose LTL-models are the TS-models of Π. For instance, (1) is splittable and its unique TS-model is captured by the unique LTL-model of the formula[1] $\neg p \wedge \Box(\neg p \leftrightarrow \bigcirc p)$.

However, splittable temporal logic programs do not cover the full expressiveness of TEL. A simple example of non-splittable formula is $\Diamond p$ (read as p "eventually holds") which can be seen as the infinitary ASP disjunction $p(0) \vee p(1) \vee p(2) \vee \ldots$ Accordingly, this formula has an infinite number of TS-models, one per each $i \geq 0$, making p true in the i-th state, and false in all the rest of states. Using several automata transformation techniques, the tool `ABSTEM` [8] allows computing the TS-models of any arbitrary temporal theory. To display the obtained TS-models, `ABSTEM` (and, in fact, `STeLP` too) uses a *Büchi automaton* [9], a finite automaton that accepts a word of *infinite length* when it corresponds to a run that visits some accepting state infinitely often. For instance, Figures 1(a) and 1(b) respectively show the Büchi automata obtained for formulas (1) and $\Diamond p$.



(a) TS-models of $\Box(\neg p \rightarrow \bigcirc p)$ (b) TS-models of $\Diamond p$

Fig. 1: A pair of Büchi automata showing the TS-models of formulas (1) and $\Diamond p$.

Until now, the question of whether the set of TS-models of any arbitrary temporal theory is LTL-representable or not was unanswered. In this way, while the Büchi automaton generated by `STeLP` is always obtained from an LTL formula, in the case of

[1] Note how this formula can be seen as a kind of "temporal completion" of (1) where implication is completed with double implication.

ABSTEM, that accepts arbitrary theories, the existence of such a formula was not guaranteed. It is true, however, that in simple cases of arbitrary theories, that formula can usually be found by *ad hoc* inspection. For instance, it is not difficult to see that the TS-models captured by the automaton 1(b) corresponding to $\Diamond p$ are exactly the LTL-models of $\neg p\, \mathcal{U}\ (p \wedge \bigcirc\Box\neg p)$ (where '$\mathcal{U}$' is the "until" LTL-operator). It was natural, therefore, to wonder whether this formula exists in the general case, that is, whether TS-models of any theory are LTL-representable.

It is perhaps worth to mention a pair of aspects that unveil the theoretical relevance of this question. First, it is well-known that Büchi automaton are, in general, more expressive than LTL. A simple example showing this limitation of LTL is the incapability of expressing the well-known *even-state* property, that consists in fixing some proposition p true in all even states, leaving it free for odd states[2], as shown by the simple Büchi automaton in Figure 2. In this way, we ignored whether there existed some theories for which their TS-models were in the more expressive set of languages included in the Büchi family but not in LTL. In particular, we ignored whether the even-state property was representable in TEL, for instance.

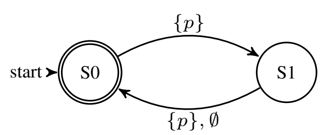


Fig. 2: A Büchi automaton capturing the even-state property.

The second important observation is that not all variants of Equilibrium Logic satisfy an analogous feature. While in the propositional case of Equilibrium Logic, it is known that the stable models of an arbitrary propositional theory can be captured by a classical propositional formula (for instance, applying loop formulas as in [10]), when one considers the first-order extension, this is not true any more. To be precise, there are first-order theories for which their stable models cannot be captured by a classical first-order formula. A clear example is the representation of the reachability property in a graph, which can be easily expressed in equilibrium logic using a recursive predicate definition, whereas it constitutes a well-known example of non-representable property in classical first-order logic.

In this paper we prove indeed that the TS-models of any temporal theory are LTL-representable. The proof for this result is challenging in the sense that involves dealing with formal background from two different areas. On the temporal logic side, it relies on the classical *Kamp's theorem* [4], that shows that LTL is exactly as expressive as Monadic First Order Logic with a linear ordering relation, MFO($<$). In particular, we use a recent alternative proof by Rabinovich [11] showing that MFO($<$) is reducible to a kind of decomposition formulas [12] called $\overrightarrow{\exists}\forall$ formulas. On the non-monotonic rea-

[2] An important assumption which is usually not stated explicitly, is that the signature only contains p or that any other proposition must vary freely in any state. Otherwise, if auxiliary atoms are allowed, the property becomes LTL-representable.

soning side, the proof uses an encoding of TEL in terms of the General Theory of Stable Models [13], a second-order characterisation of Quantified Equilibrium Logic [14], and uses some results on elimitation of second-order quantifiers from the algorithm in [15] for computing Circumscription [16].

The rest of the paper is organised as follows. In the next section, we introduce the basic definitions of TEL. In Section 3 we provide a translation from TEL, a modal formalism, to a quantified logic with monadic predicates and an linear ordering relation. We also recall the definition of first-order stable models from [13] in terms of a second-order operator. In Section 4 we recall some known results for elimination of second-order quantifiers. Section 5 contains the main proof and a small example. Finally, Section 6 concludes the paper.

2 Temporal Equilibrium Logic

The definition of TEL is made in two steps. First, we define a monotonic base, the logic of *Temporal Here-and-There* (THT), that consists in extending the monotonic base of Equilibrium Logic (called the logic of *Here-and-There* [17]) to the temporal case. Second, we define a model selection criterion among the THT-models of a temporal theory.

The syntax of propositional THT coincides with LTL. Given a set of atoms Σ, a *temporal formula* φ can be expressed by the LTL 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 α and β are temporal formulas in their turn and p is any atom. The unary temporal operator $\bigcirc$ is read as "next," and the binary temporal operators $\mathcal{U}$ and $\mathcal{R}$ are read as "until" and "release" respectively. Negation is defined as $\neg\varphi \stackrel{\text{def}}{=} \varphi \rightarrow \bot$ whereas $\top \stackrel{\text{def}}{=} \neg\bot$. As usual, $\varphi \leftrightarrow \psi$ stands for $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$. Other usual temporal operators can be defined in terms of $\mathcal{U}$ and $\mathcal{R}$. For instance, $\Box\varphi \stackrel{\text{def}}{=} \bot \,\mathcal{R}\, \varphi$ and $\Diamond\varphi \stackrel{\text{def}}{=} \top \,\mathcal{U}\, \varphi$ where $\Box$ is read "forever" and $\Diamond$ stands for "eventually" or "at some future point."

An *LTL-interpretation* (or *temporal interpretation*) $\mathbf{T}$, is an infinite sequence of sets of atoms $T_i \subseteq \Sigma$ for $i \geq 0$ called *states*. Given two temporal intepretations, $\mathbf{H}$ and $\mathbf{T}$, we define the ordering relation $\mathbf{H} \leq \mathbf{T}$ as $H_i \subseteq T_i$ for all $i \in \mathbb{N}$. A *THT-interpretation* $\mathbf{M}$ is a pair of temporal interpretations $\langle \mathbf{H}, \mathbf{T} \rangle$ such that $\mathbf{H} \leq \mathbf{T}$. When $\mathbf{H} = \mathbf{T}$ the THT-interpretation is said to be *total*.

Definition 2 (THT satisfaction). *Given an interpretation* $\mathbf{M} = \langle \mathbf{H}, \mathbf{T} \rangle$ *and a state number* $i \in \mathbb{N}$, *we recursively define when* $\mathbf{M}$ satisfies *a temporal formula* φ *at* i *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* $\exists j \in \{i, \ldots, k\text{-}1\}, \mathbf{M}, j \models \varphi$.

■

From the definition of $\Box\varphi$ and $\Diamond\varphi$ as derived operators, we can easily obtain:

- $\mathbf{M}, i \models \Box\varphi$ iff $\forall k \geq i$, $\mathbf{M}, k \models \varphi$
- $\mathbf{M}, i \models \Diamond\varphi$ iff $\exists k \geq i$ $\mathbf{M}, k \models \varphi$

A formula φ is THT-*valid* if $\mathbf{M}, 0 \models \varphi$ for any $\mathbf{M}$. An interpretation $\mathbf{M}$ is a THT-*model* of a theory C, written $\mathbf{M} \models C$, if $\mathbf{M}, 0 \models \varphi$, for all formula $\varphi \in C$.

It is not difficult to see that, when we restrict to total interpretations, $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \varphi$ in THT iff $\mathbf{T}, i \models \varphi$ in LTL. In other words, total THT-models correspond to LTL-models. We establish next a model selection criterion for total THT-models.

Definition 3 (Temporal Equilibrium Model). *A total THT-model* $\mathbf{M} = \langle \mathbf{T}, \mathbf{T} \rangle$ *is a* Temporal Equilibrium Model *of a theory* C *iff there is no other temporal interpretation* $\mathbf{H}$ *such that* $\mathbf{H} < \mathbf{T}$ *and* $\langle \mathbf{H}, \mathbf{T} \rangle, 0 \models C$. ■

We say that $\mathbf{T}$ is a *temporal stable model* (*TS-model* for short) of a theory C iff $\langle \mathbf{T}, \mathbf{T} \rangle$ is a temporal equilibrium model of C. Since temporal equilibrium models are total THT-models, any TS-model is an LTL-model.

3 From Modal Logic to Quantified Predicates

The classical main result from Kamp's dissertation [4] consisted in proving that LTL has the same expressive power than Monadic First-Order logic with a linear ordering relation, MFO($<$). In this section, we prove that one of the two directions of Kamp's proof, the fact that LTL can be translated to MFO($<$), can be actually applied both to THT and TEL using the same translation.

We will consider formulas in the standard syntax of first-order logic with no functions, with a binary predicate '$\leq$' representing a linear ordering relation and an arbitrary set $\mathcal{P}$ of monadic predicates. We will use letters F, G to denote first-order formulas, as opposed to Greek letters φ, ψ to denote temporal formulas. We keep the definitions of derived operators '$\neg$', '$\top$' and '$\leftrightarrow$' as before and additionally define $i < j \stackrel{\text{def}}{=} \neg(j \leq i)$, $i = j \stackrel{\text{def}}{=} (i \leq j) \wedge (j \leq i)$. Although functions are not allowed, we will use the notation $p(t+1)$ as an abbreviation of the formula $\exists x\ (p(x) \wedge t \leq x \wedge \neg \exists y\ (t < y \wedge y < x))$, for any monadic predicate p. The domain will always be fixed to the set $\mathbb{N}$ of natural numbers[3] and we will deal with a syntactic constant i per each element $i \in \mathbb{N}$. Given a set of monadic predicates $\mathcal{P}$ we denote $Atoms(\mathcal{P})$ as the set of all possible atoms formed by predicates in $\mathcal{P}$ and constants in $\mathbb{N}$. Since the domain and the interpretation of $\leq$ will be fixed, a first-order interpretation can be succinctly represented as a set of atoms $\mathcal{T} \subseteq Atoms(\mathcal{P})$. There exists an obvious one-to-one correspondence between a first-order interpretation $\mathcal{T}$ defined in this way and a temporal interpretation $\mathbf{T} = \{T_i\}_{i \in \mathbb{N}}$ so that $p \in T_i$ iff $p(i) \in \mathcal{T}$ for any $i \in \mathbb{N}$.

[3] Although the original Kamp's result holds for any time model in the form of a Dedekind complete linear ordering, for our purposes, it suffices with considering the ordered set of natural numbers.

We define next the logic of *Monadic Quantified Here-and-There* with a linear ordering, MHT($<$), as a simplified instance of [14]. An MHT($<$)-*interpretation* is a tuple $\mathcal{M} = \langle \mathcal{H}, \mathcal{T} \rangle$ where $\mathcal{H} \subseteq \mathcal{T} \subseteq Atoms(\mathcal{P})$. As before, we say that the interpretation is *total* iff $\mathcal{H} = \mathcal{T}$. The satisfiability relation is defined as follows:

- $\mathcal{M} \models p(i)$ iff $p(i) \in \mathcal{H}$
- $\mathcal{M} \models i \leq j$ iff i is less or equal than j as natural numbers
- $\wedge$, $\vee$, $\perp$ as usual
- $\mathcal{M} \models F \rightarrow G$ iff for all $w \in \{\mathcal{H}, \mathcal{T}\}$, $\langle w, \mathcal{T} \rangle \not\models F$ or $\langle w, \mathcal{T} \rangle \models G$
- $\mathcal{M} \models \forall x\ F(x)$ iff for all $i \in \mathbb{N}$, $\mathcal{M} \models F(i)$
- $\mathcal{M} \models \exists x\ F(x)$ iff there exists $i \in \mathbb{N}$, $\mathcal{M} \models F(i)$

THT formulas can be regarded as a fragment of MHT($<$) where only one free variable is considered[4], as happens when encoding LTL into MFO($<$). We introduce next a simple syntactic translation of a temporal formula into a first-order formula.

Definition 4. *Let φ be a temporal formula for signature Σ. We define the translation $[\varphi]_t$ for some term t as follows:*

$$
\begin{aligned}
[\perp]_t &\stackrel{\text{def}}{=} \perp \\
[p]_t &\stackrel{\text{def}}{=} p(t), \text{ \textit{with} } p \in \Sigma \text{ \textit{an atom.}} \\
[\alpha \wedge \beta]_t &\stackrel{\text{def}}{=} [\alpha]_t \wedge [\beta]_t \\
[\alpha \vee \beta]_t &\stackrel{\text{def}}{=} [\alpha]_t \vee [\beta]_t \\
[\alpha \rightarrow \beta]_t &\stackrel{\text{def}}{=} [\alpha]_t \rightarrow [\beta]_t \\
[\bigcirc \alpha]_t &\stackrel{\text{def}}{=} [\alpha]_{t+1} \\
[\alpha\, \mathcal{U}\, \beta]_t &\stackrel{\text{def}}{=} \exists x\ (t \leq x \wedge [\beta]_x \wedge \forall y\ (t \leq y < x \rightarrow [\alpha]_y)) \\
[\alpha\, \mathcal{R}\, \beta]_t &\stackrel{\text{def}}{=} \forall x\ (t \leq x \rightarrow [\beta]_x \vee \exists y\ (t \leq y < x \wedge [\alpha]_y))
\end{aligned}
$$

Note how, per each atom $p \in \Sigma$ 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:

$$[\Box \alpha]_t \stackrel{\text{def}}{=} \forall x\ (t \leq x \rightarrow [\alpha]_t) \qquad [\Diamond \alpha]_t \stackrel{\text{def}}{=} \exists x\ (t \leq x \wedge [\alpha]_t)$$

As a pair of examples, the translations of our two running examples (1) and $\Diamond p$ for $t = 0$ respectively correspond to:

$$\forall x\ (0 \leq x \ \rightarrow (\neg p(x) \rightarrow p(x+1))) \tag{2}$$

$$\exists x (0 \leq x \wedge p(x)) \tag{3}$$

The trivial direction of Kamp's theorem guarantees that, given any temporal formula φ, there exists an obvious one-to-one correspondence between LTL-models of φ and MFO($<$)-models of $[\varphi]_0$. We show next that this correspondence is maintained for THT-interpretations and MHT($<$)-models.

[4] Up to date, the question of whether the other direction of Kamp's theorem holds for THT or not is unanswered. Namely, we ignore whether MHT($<$) can be translated back to THT.

Theorem 1. *Let φ be a temporal formula for a vocabulary Σ. Moreover, let $\mathbf{M} = \langle \mathbf{H}, \mathbf{T} \rangle$ be a THT-interpretation for Σ and let $\mathcal{M} = \langle \mathcal{H}, \mathcal{T} \rangle$ be its corresponding MHT(<)-interpretation. Then $\mathbf{M}, i \models \varphi$ in THT iff $\mathcal{M} \models [\varphi]_i$ in MHT(<).*

Proof. It directly follows from structural induction (see Appendix for full detail). ■

3.1 First-order Stable Models

We say that a total MHT(<)-model $\mathcal{M} = \langle \mathcal{T}, \mathcal{T} \rangle$ of a first-order theory C is an *equilibrium model* of C iff there is no strictly smaller $\mathcal{H} \subset \mathcal{T}$ such that $\langle \mathcal{H}, \mathcal{T} \rangle$ is a model of C. We call *stable model* of C to any $\mathcal{T}$ such that $\langle \mathcal{T}, \mathcal{T} \rangle$ is an equilibrium model of C.

We show next that the correspondence shown for the previous translation is still valid when we consider TS-models versus (first-order) stable models.

Theorem 2. *Let $\mathbf{T}$ be a temporal interpretation, $\mathcal{T}$ its corresponding first-order interpretation and φ some temporal formula. Then, $\mathbf{T}$ is a TS-model of φ iff $\mathcal{T}$ is a stable model of $[\varphi]_0$.*

Proof. We begin noting that the ordering relation between temporal interpretations $\mathbf{H} \leq \mathbf{T}$ is also in one-to-one correspondence with the ordering relation $\mathcal{H} \subseteq \mathcal{T}$ as sets of monadic predicate atoms. Then, the result follows from the definitions of TS-model and stable model, together with Theorem 1. ■

We conclude this section recalling an important characterisation of stable models for first-order theories in terms of second-order logic introduced in [13] that will be crucial for proving the main result. We define next the SM operator from [13] slightly adapted for our particular kind of first-order theories.

Definition 5 (SM Operator (adapted from [13])). *Let $\overline{p}$ be the list of monadic predicate constants $p_1 \cdots p_n$ in $\mathcal{P}$. For any first-order formula F we define:*

$$SM[F;\overline{p}] \stackrel{\text{def}}{=} F \wedge \neg\exists\overline{u}\ (\overline{u} \prec \overline{p} \wedge F^*(\overline{u}))$$

where $\overline{u}$ is a list of n distinct predicate variables $u_1 \cdots u_n$, $\overline{u} \prec \overline{p}$ stands for the conjunction of all the formulas

$$\forall x\ (u_i(x) \to p_i(x)) \wedge \exists y\ (p_i(y) \wedge \neg u_i(y))$$

for $i = 1, \ldots n$, and $F^*(\overline{u})$ is a recursive transformation defined as:

$$\begin{aligned}
p_i(t)^* &\stackrel{\text{def}}{=} u_i(t) \text{ for any term } t \text{ and } p_i \text{ in } \overline{p} \\
(t \leq t')^* &\stackrel{\text{def}}{=} t \leq t' \\
(F \wedge G)^* &\stackrel{\text{def}}{=} F^* \wedge G^* \\
(F \vee G)^* &\stackrel{\text{def}}{=} F^* \vee G^* \\
(F \to G)^* &\stackrel{\text{def}}{=} (F \to G) \wedge (F^* \to G^*) \\
(\forall x F)^* &\stackrel{\text{def}}{=} \forall x F^* \\
(\exists x F)^* &\stackrel{\text{def}}{=} \exists x F^*
\end{aligned}$$

Theorem 3 (from [13]). *Given a first-order formula F for predicates $\overline{p}$, any interpretation $\mathcal{M}$ is a model of SM$[F;\overline{p}]$ iff $\mathcal{M}$ is a stable model (in the sense of Equilibrium Logic) of F.* ∎

4 Second-Order Quantifier Elimination

Second order logic has been successfully used in non-monotonic formalisms such as Circumscription [16,18] and General Stable Models [13]. Both formalisms are expressed in terms of a second order formula that sometimes can be reduced to a first-order expression. This reduction can be syntactically performed by means of combining some well-known equivalences on second-order logic together with Ackermann's Lemma [19]. Both results are described below.

Proposition 1. *The following pairs of formulas are equivalent in second-order logic.*

$$\neg\neg A \equiv A \tag{4}$$
$$\neg(A \wedge B) \equiv \neg A \vee \neg B \tag{5}$$
$$\neg(A \vee B) \equiv \neg A \wedge \neg B \tag{6}$$
$$\neg\forall x A(x) \equiv \exists x \neg A(x) \tag{7}$$
$$\neg\exists x A(x) \equiv \forall x \neg A(x) \tag{8}$$
$$\exists x A(x) \vee B(x) \equiv \exists x A(x) \vee \exists x B(x) \tag{9}$$
$$\forall x A(x) \wedge B(x) \equiv \forall x A(x) \wedge \forall x B(x) \tag{10}$$
$$Qx(A(x)) \wedge C \equiv Qx(A(x) \wedge C) \tag{11}$$
$$C \wedge Q\ x(A(x)) \equiv Qx(C \wedge A(x)) \tag{12}$$
$$Qx(A(x)) \vee C \equiv Qx(A(x) \vee C) \tag{13}$$
$$C \vee Q\ x(A(x)) \equiv Qx(C \vee A(x)) \tag{14}$$
$$QxQyA \equiv QyQxA \tag{15}$$
$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C) \tag{16}$$
$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C) \tag{17}$$
$$A(\overline{t}) \equiv \forall \overline{x}\ (\overline{x} = \overline{t}) \rightarrow A(\overline{x}) \tag{18}$$

$$A(\overline{t_1}) \vee \cdots \vee A(\overline{t_n}) \equiv \exists \overline{x}\ (\overline{x} = \overline{t_1} \vee \cdots \vee \overline{x} = \overline{t_n}) \wedge A(\overline{x}) \tag{19}$$
$$\forall \overline{x} \exists y A(\overline{x}, \cdots) \equiv \exists f \forall \overline{x} A(\overline{x}, f(\overline{x}), \cdots) \tag{20}$$
$$A(\overline{t_1}) \wedge \cdots A(\overline{t_n}) \equiv \forall \overline{x}(\overline{x} \neq \overline{t_1} \wedge \cdots \overline{x} \neq \overline{t_n}) \vee A(\overline{x}) \tag{21}$$

Here Q stands for any quantifier and A, B, C are formulas such that C does not contain free occurrences of the variable x. In clauses (18), (19) *and* (21), $\overline{t}$, $\overline{t_1}$, $\cdots$, $\overline{t_n}$ *contain variables from* $\overline{x}$. *In clause* (20), f *is a function variable that does not occur in A.* ∎

This list of syntactic equivalences allows us to transform an input second-order formula into an equivalent one that has a suitable form for applying the Ackermann's Lemma, which is following defined:

Lemma 1 (Ackermann's Lemma). *Let Φ be a predicate variable and $G(\overline{x}, \overline{z})$, $B(\Phi)$ formulas without second order quantification such that $G(\overline{x}, \overline{z})$ does not contain Φ . The following equivalences hold:*

$$\exists \Phi \forall \overline{x}(\Phi(\overline{x}) \to G(\overline{x}, \overline{z})) \wedge B(\Phi) \equiv B(\Phi \leftarrow G(\overline{x}, \overline{z})) \text{ If } B(\Phi) \text{ is positive for } \Phi$$
$$\exists \Phi \forall \overline{x}(G(\overline{x}, \overline{z}) \to \Phi(\Phi)) \wedge B(\Phi) \equiv B(\Phi \leftarrow G(\overline{x}, \overline{z})) \text{ If } B(\Phi) \text{ is negative for } \Phi$$

where, in the right-hand formulas, the arguments $\overline{x}$ of G are each time substituted by the respective actual arguments of Φ (renaming the bound variables whenever necessary).

The process of removing second-order quantifiers has been automated in two algorithms: SCAN [20] and DLS [15]. While SCAN is applicable to any arbitrary second-order formula but may not terminate, DLS provides an input format for which the algorithm terminates removing the second-order quantifiers successfully. This input format is defined below:

Lemma 2 (From [15]). *Let $\exists \Phi\, [(pref\ B) \wedge (pref'\ C)]$ where $pref$ and $pref'$ are sequences of first-order quantifiers; B and C are quantifier-free formulas in conjunctive normal form; B is positive w.r.t. Φ and C is negative w.r.t. Φ then the DLS algorithm will remove all second-order quantifiers iff one of the following conditions holds:*

- *B is universal and each conjunct of B contains at most one occurrence of Φ, or*
- *C is universal and each conjunct of C contains at most one occurrence of $\neg\Phi$* ■

By positive occurrence (resp. negative) of a predicate variable Φ in a formula φ we mean that the conjunctive normal form of φ contains a subformula of the form $\Phi(\overline{t})$ (resp. $\neg\Phi(\overline{t})$)). A formula φ is said to be *positive* (resp. *negative*) w.r.t. Φ iff all occurrences of Φ in φ are positive (resp. negative).

5 Representing TS-models in LTL

In this section we prove that the set of TS-models of a temporal formula φ is always LTL-representable. To achieve this result we proceed by proving that the set of stable models of $[\varphi]_0$ collapses to a first order sentence. The kind of second-order formulas that are considered in this paper has the form $\exists \Phi\ F(\Phi)$ where Φ is a predicate name and F a monadic first-order formula. We will resort to a recent result proved by Rabinovich [11] showing that we can replace F by an equivalent[5] disjunction of the so called $\overrightarrow{\exists}\forall$ formulas which are described below:

Definition 6. *An $\overrightarrow{\exists}\forall$ formula over a set of monadic predicates $\mathcal{P}$ is a has the form:*

[5] This holds for any $\geq$ representing a Dedekind complete chains.

$$
\begin{aligned}
F(z_0, \cdots z_n) := \exists x_n \cdots \exists x_1 \exists x_0 \cdot & \\
& \left(\bigwedge_{k=0}^{m} z_k = x_{i_k} \right) \wedge (x_n > x_{n-1} > \cdots > x_1 > x_0) \\
& \wedge \left(\bigwedge_{j=0}^{n} A_j(x_j) \right) \wedge \left(\bigwedge_{j=1}^{n} [\forall y \; (x_{j-1} < y < x_j) \to B_j(y)] \right) \\
& \wedge \forall y \; (x_n < y) \to B_{n+1}(y)) \;\wedge\; \forall y \; (y < x_0) \to B_0(y) \qquad (22)
\end{aligned}
$$

with a prefix of n+1 existential quantifiers and with all A_j, B_j quantifier free formulas with one variable over $\mathcal{P}$, and $i_0, \cdots, i_m \in \{0, \cdots, n\}$.

In his paper [11], Rabinovich uses this normal form (disjunction of $\overrightarrow{\exists}\forall$ formulas) to prove the non-trivial direction of Kamp's theorem we enunciate below, proved both in [4] and [11] (and many other alternatives in the literature).

Theorem 4 (Kamp's theorem). *For every MFO(<) formula $F(x)$ with one free variable, there is an LTL formula which is equivalent to $F(x)$ over Dedekind complete chains.* ■

Since any MFO(<) formula F can be reduced to a disjunction of $\overrightarrow{\exists}\forall$ formulas, and since existential quantifiers distribute with respect to disjunction, the problem of removing the second-order existential quantifier in $\exists\Phi \; F(\Phi)$ can be exclusively focused on expressions of the form $\exists\Phi \; G(\Phi)$ where G is a $\overrightarrow{\exists}\forall$ formula.

Lemma 3. *Let $\mathcal{P}$ be a set of predicate names, Φ a predicate variable and $F(\Phi)$ an $\overrightarrow{\exists}\forall$ formula built over $\mathcal{P}$. Then, the formula $\exists\Phi \; F(\Phi)$ is equivalent to a first-order formula. NOTE: this property is FALSE[6]. As a counterexample, take the $MSO(<)$ formula $\exists\Phi F(\Phi)$ with:*

$$
F(\Phi) := \Phi(0) \wedge \neg\Phi(1) \wedge \forall x \geq 0 \cdot (\Phi(x) \leftrightarrow \Phi(x+2)) \wedge \forall x \geq 0 \cdot (\Phi(x) \to p(x))
$$

Intuitively, $\Phi(x)$ is true if and only if state x is even. As a result of the last implication, predicate $p(x)$ is true in even states and varies freely in all the rest. This is known as the "even-state property" and it is well-known to be non-representable in LTL. Now, if the second order variable Φ could be removed, it would mean that the resulting formula $G(p)$ would be an $FSO(<)$ expressing the even-state property for $p(x)$. But, by Kamp's theorem, there would be an equivalent LTL formula $G'(p)$ expressing the same property and we reach a contradiction.

Proof. By Definition 6, $F(\Phi)$ has the form (22) where Φ will only occur in subformulas A_j, B_j, B_0 or B_{n+1}, since all the rest is expressed in terms of the $\leq$ predicate. Taking

[6] We wish to thank Stéphane Demri for finding both the counterexample and the error in the proof.

the formula $\exists\Phi F(\Phi)$, we can shift the quantifier $\exists\Phi$ inside (22) leaving outside all those subformulas that do not depend on Φ as follows:

$$\begin{aligned}\exists x_n \cdots \exists x_1 \exists x_0 \cdot \; & \left(\bigwedge_{k=0}^{m} z_k = x_{i_k}\right) \wedge (x_n > x_{n-1} > \cdots > x_1 > x_0) \\ & \wedge \exists\Phi \Bigg(\textstyle\bigwedge_{j=0}^{n} A_j(x_j) \;\wedge\; \bigwedge_{j=1}^{n} [\forall y\, (x_{j-1} < y < x_j) \to B_j(y)] \\ & \qquad \wedge [\forall y\, (x_n < y) \to B_{n+1}(y)] \;\wedge\; [\forall y\, (y < x_0) \to B_0(y)] \Bigg)\end{aligned} \tag{23}$$

Now, let us take the subformula G above in the scope of $\exists\Phi$ and transform it into CNF:

$$\begin{aligned} & \bigwedge_{j=0}^{n} \bigwedge_{k=1}^{m_j} A_{j,k}(x_j) \\ \wedge & \left[\bigwedge_{j=1}^{n} \forall y \left[\bigwedge_{k=1}^{r_j} (\neg(x_{j-1} < y < x_j) \vee B_{j,k}(y))\right]\right] \\ \wedge\; & \forall y \left[\bigwedge_{k=1}^{h} (\neg(x_n < y) \vee B_{n+1,k}(y))\right] \\ \wedge\; & \forall y \left[\bigwedge_{k=1}^{s} (\neg(y < x_0) \vee B_{0,k}(y))\right] \end{aligned}$$

We can shift $\forall y$ outside using (10), (11) and (12) and reorganise the clauses as follows:

$$\begin{aligned}\forall y \Bigg(\quad & \left[\textstyle\bigwedge_{j=0}^{n} [\bigwedge_{k=1}^{m_j} A_{j,k}(x_j) \wedge \bigwedge_{k=1}^{r_j} (\neg(x_{j-1} < y < x_j) \vee B_{j,k}(y))]\right] \\ \wedge\; & \left[\textstyle\bigwedge_{k=1}^{h} (\neg(x_n < y) \vee B_{n+1,k}(y))\right] \wedge\; [\textstyle\bigwedge_{k=1}^{s} (\neg(y < x_0) \vee B_{0,k}(y))] \Bigg)\end{aligned} \tag{24}$$

The resulting formula (24) is universally quantified (NOTE: a first part of the error comes here: the formula has some *free* variables x_i that, in fact, are externally quantified with $\exists$) and in prefex normal form, while its matrix is in CNF and Φ occurring at most once in each conjunct. It is crucial to note that any of these conjuncts containing both positive and negative occurrences of Φ can be removed. This is because all subformulas that may contain Φ, that is, $A_{j,k}(x_j)$, $B_{j,k}(y)$, $B_{+1,k}(y)$ and $B_{0,k}(y)$, only contain one free variable, y, and so, the positive and negative occurrence of Φ in a CNF clause of this form mandatorily forms a disjunction $\Phi(y) \vee \neg\Phi(y)$, which is a tautology. Since (24) is equivalent to G, the quantified formula $\exists\Phi G$ in (23) can be reformulated as:

$$\bigwedge_{j=0}^{h'} C'_j(y) \wedge \;\exists\Phi \left(\forall y \left[\left[\bigwedge_{j=0}^{h^+} C_j^+(y)\right] \wedge \left[\bigwedge_{j=0}^{h^-} C_j^-(y)\right]\right]\right)$$

where C_j^+ and C_j^- represent the conjuncts which contain positive and negative occurrences of Φ whereas C_j' stands for all conjuncts that do not depend on Φ, and so, have been moved out of the scope of $\exists\Phi$. Finally, the expression above is already in the input form for the DLS algorithm, so the removal of the second order quantifier is guaranteed by Lemma **??** (NOTE: This is wrong. DLS algorithm is only applicable if *all* first-order variables are universally quantified. In this case, we have some variables x_i that are free and come from an existential quantification in the initial formula). ■

As said before, this lemma can be easily extended to the case of a disjunction of $\vec{\exists}\forall$ formulas by distributing the existential quantifier with respect to disjunction and then applying Lemma 3 to every disjunct.

Theorem 5 (Main theorem). *The set of TS-models of a temporal formula φ is LTL-representable.*

Proof. From Theorem 2, we know that the (first-order) stable models of $[\varphi]_0 = F$ are in one-to-one correspondence with the TS-models of φ. Those first-order stable models are captured by the second-order formula $\text{SM}[F;\overline{p}] = F \wedge \neg\exists\overline{u}\ (\overline{u} \prec \overline{p} \wedge F^*(\overline{u}))$. Now, as $F^*(\overline{u})$ is an MFO($<$) formula, it can be expressed as a disjunction of $\vec{\exists}\forall$ formulas. By Lemma 3, we can go applying algorithm DLS to remove one by one all second-order quantifiers $\exists u_i$ from right to left to eventually obtain a first-order formula equivalent to $\text{SM}[F;\overline{p}]$. But then, by Kamp's theorem, this formula, which is in the syntax of MFO($<$), can be expressed back in LTL. ■

5.1 A pair of examples

In this section we show how to obtain LTL formulas that capture the TS-models for our two running examples (1) and $\Diamond p$. Theorem 5 guarantees that this is always possible, but the method to obtain such formulas used in its proof resorts to the DLS algorithm and to the non-trivial direction of Kamp's theorem. In both cases, this implies a rather complicated set of transformations we did not include in this paper, for simplicity sake. Instead, we will use Ackermann's Lemma for second-order quantifier elimination and will reduce the obtained MFO($<$) formulas back to LTL by hand.

Example 1. Take the (non-splittable) temporal formula $\Diamond p$ and let us call F to the formula $[\Diamond p]_0 =$ (3). In its turn, the stable models of (3) are captured by:

$$\text{SM}[(3);p] \equiv (3) \wedge \underbrace{\neg\exists u\ \ (u \prec p \wedge \exists z\ \ (z \geq 0 \wedge u(z)))}_{F}$$

Let us call F to the second conjunct. Using the definition of $u \prec p$, F is equivalent to:

$$F \equiv \neg\exists u \left(\forall y\ \ (u(y) \rightarrow p(y)) \wedge \exists k\ \ (p(k) \wedge \neg u(k)) \wedge \exists z\ \ (z \geq 0 \wedge u(z))\right)$$

We can move outside the scope of $\exists u$ the first-order existential quantifiers $\exists k$ and $\exists z$ and, afterwards, the conjuncts that do not depend on u, $p(k)$ and $z \geq 0$. We also use equivalence (18) on $u(z)$ to obtain:

$$\equiv \neg\exists k\ \exists z\Big(z \geq 0 \wedge p(k) \wedge \exists u\Big(\neg u(k) \wedge \forall y\,(u(y) \rightarrow p(y)) \wedge \forall h\,(h = z \rightarrow u(h))\Big)\Big)$$

At this point, we can apply Ackermann's lemma taking $G = (h = z)$ and $H = \neg u(k) \wedge \forall y\ (p(y) \rightarrow u(y))$. The resulting formula is

$$\begin{aligned}
&\equiv \neg\exists k\ \exists z\ \Big(z \geq 0 \wedge p(k) \wedge k \neq z \wedge \forall y\ \ (y = z \rightarrow p(y))\Big)\\
&\equiv \neg\exists k\ \exists z\ \Big(z \geq 0 \wedge p(k) \wedge k \neq z \wedge p(z)\Big)\\
&\equiv \forall k\ \forall z\ (z \geq 0 \wedge p(z) \wedge p(k) \rightarrow k = z)
\end{aligned}$$

This formula essentially expresses that there is at most one point k with $p(k)$ true. Thus, the complete final result for SM$[(3); p]$ is:

$$\text{SM}[(3); p] \equiv \exists x(x \geq p \wedge p(x)) \wedge \forall k\ \forall z\ (z \geq 0 \wedge p(z) \wedge p(k) \rightarrow k = z)$$

that means that there is exactly one point with $p(x)$ true. This can be reexpressed as:

$$\begin{aligned}
&\equiv \exists x\Big(x \geq 0 \wedge p(x) \wedge \forall z(z \geq 0 \wedge z \neq x \rightarrow \neg p(z))\Big)\\
&\equiv \exists x\Big(x \geq 0 \wedge p(x) \wedge \forall z(z \geq 0 \wedge z < x \rightarrow \neg p(z)) \wedge \forall z(z \geq 0 \wedge z > x \rightarrow \neg p(z))\Big)\\
&\equiv \exists x\Big(x \geq 0 \wedge p(x) \wedge \forall z(0 \leq z < x \rightarrow \neg p(z)) \wedge \forall z(z \geq x + 1 \rightarrow \neg p(z))\Big)
\end{aligned}$$

Finally, it can be easily checked that this last expression corresponds to the translation of the LTL formula $\neg p\,\mathcal{U}\ (p \wedge \bigcirc\Box\neg p)$. ■

Example 2. Take the formula (1) and its translation $[(1)] = (2)$. Its corresponding stable models are captured by

$$\text{SM}[(2); p] = (2) \wedge \underbrace{\neg\exists u\ \Big(u \prec p \wedge \forall x\ (0 \leq x \wedge \neg p(x) \rightarrow u(x + 1))\Big)}_{F}$$

Let us focus on the second conjunct, F. By applying equivalences (15), (18) and some simple transformations we get:

$$F \equiv \quad \neg\exists u\ \Big(u \prec p \wedge \forall k\ (\exists x\ \ (0 \leq x \wedge \neg p(x) \wedge k = x + 1) \rightarrow u(k))\Big)$$

Since $u \prec p$ only contains negative occurrences of u we can easily apply Ackermann's lemma considering $B(u) = u \prec p = \forall y(u(y) \rightarrow p(y)) \wedge \exists z(p(z) \wedge \neg u(z))$ and

$G(k) = \exists x \ \ (0 \leq x \wedge \neg p(x) \wedge k = x+1)$. As a result, we get the equivalent first-order formula:

$$\equiv \neg \Bigg(\overbrace{\forall y \ \ (\exists x \ \ (0 \leq x \wedge \neg p(x) \wedge y = x+1) \rightarrow p(y))}^{C}$$
$$\wedge \exists z \, (p(z) \wedge \neg \exists x \ \ (0 \leq x \wedge \neg p(x) \wedge z = x+1)) \Bigg)$$

Now, for C, we can separate $y = 0$ from $y > 0$. If $y = 0$ there is no $x \geq 0$ such that $y = x + 1$ so the implication becomes trivially true. For $y > 0$ we can reexpress C as $\forall y \, (y > 0 \wedge \neg p(y-1) \rightarrow p(y))$ or just $\forall y \, (y \geq 0 \wedge \neg p(y) \rightarrow p(y+1))$ which is equivalent to (2). Therefore:

$$\begin{aligned} \mathrm{SM}[(2); p] &\equiv (2) \wedge \Bigg(\neg(2) \vee \neg \Bigg(\exists z \, (p(z) \wedge \neg \exists x \ \ (0 \leq x \wedge \neg p(x) \wedge z = x+1)) \Bigg) \Bigg) \\ &\equiv (2) \wedge \forall z \Bigg(\neg \exists x \ \ (0 \leq x \wedge \neg p(x) \wedge z = x+1) \rightarrow \neg p(z) \Bigg) \end{aligned}$$

We apply a similar technique as before, separating $z = 0$ from $z > 0$ to obtain:

$$\begin{aligned} &\equiv (2) \wedge \neg p(0) \wedge \forall z (z \geq 0 \wedge p(z) \rightarrow \neg p(z+1)) \\ &\equiv \forall x (x \geq 0 \wedge \neg p(x) \rightarrow p(x+1)) \wedge \neg p(0) \wedge \forall z \Bigg(z \geq 0 \wedge p(z) \rightarrow \neg p(z+1) \Bigg) \\ &\equiv \forall x \Bigg(x \geq 0 \rightarrow (\neg p(x) \leftrightarrow p(x+1)) \Bigg) \end{aligned}$$

which obviously corresponds to the LTL formula $\neg p \wedge \Box(\neg p \leftrightarrow \bigcirc p)$. ■

6 Conclusions

In this paper we have shown that the temporal stable models (TS-models) of an arbitrary formula in the syntax of Linear-Time Temporal Logic (LTL) can always be captured as the LTL models of another temporal formula. Until now, we only knew that the set of TS-models of a formula could always be captured by a Büchi automaton, but in the general case, the latter is more expressive than LTL. For instance, we know now that properties that are Büchi-representable but not LTL-representable (such as the even-state property) cannot be expressed as the TS-models of any theory. Therefore, this result confirms that the non-monotonicity of TEL does not increase the expressive power we obtain with respect to monotonic LTL, although it provides a more flexible and elaboration tolerant representation.

The interest of the obtained result is mostly theoretical. From a practical point of view, we may find the Büchi automaton representation for TS-models more comfortable and readable, especially in small examples. Still, there may be cases in which an LTL-formula ψ can provide a more compact description of the set of TS-models of another formula φ. For instance, for skeptical reasoning, checking if some query α holds in all TS-models of φ amounts to checking if the formula $\psi \wedge \neg\alpha$ is LTL-unsatisfiable. For credulous reasoning, checking whether α holds in some TS-model would correspond to checking the LTL-satisfiability of $\psi \wedge \alpha$.

The obtained proof has resulted challenging in the sense that required dealing with technical results both from the LTL arena and from logical encodings of first-order stable models, including techniques for elimination of second-order quantifiers. Although the main theorem in this paper shows that there always exists an LTL-formula ψ that captures the TS-models of any temporal formula φ, the proof itself does not provide a direct automated method for obtaining ψ, since it relies on previous results of expressiveness whose practical applicability is not always straightforward. The construction of an automated method for that purpose is part of the immediate future work.

Acknowledgements We are especially thankful to Vladimir Lifschitz who suggested us using the results from [15] for second-order quantifier elimination.

References

1. 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
2. Pearce, D.: A new logical characterisation of stable models and answer sets. In: NMELP'96. Volume 1216 of Lecture Notes in Artificial Intelligence. Springer (1996) 57–70
3. Pearce, D.: Equilibrium logic. Annals of Mathematics and Artificial Intelligence **47**(1–2) (2006) 3–41
4. Kamp, J.: Tense Logic and the Theory of Linear Order. PhD thesis, University of California at Los Angeles (1968)
5. Manna, Z., Pnueli, A.: The Temporal Logic of Reactive and Concurrent Systems: Specification. Springer-Verlag (1991)
6. 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
7. Aguado, F., Cabalar, P., Pérez, G., Vidal, C.: Loop formulas for splitable temporal logic programs. In: LPNMR'11. Volume 6645 of Lecture Notes in Computer Science. (2011) 80–92
8. 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). (2014)
9. Büchi, R.: On a decision method in restricted second-order arithmetic. In: Intl. Congress on Logic, Method and Philosophical Science'60. (1962) 1–11
10. Lee, J., Lifschitz, V.: Loop formulas for disjunctive logic programs. In: Proceedings of the 19th International Conference on Logic Programming (ICLP'03). (2003) 451–465
11. Rabinovich, A.: A proof of kamp's theorem. In: CSL. (2012) 516–527
12. Gabbay, D., Pnueli, A., Shelah, S., Stavi, J.: On the temporal analysis of fairness. In: Proceedings of the 7th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. POPL '80, New York, NY, USA, ACM (1980) 163–173
13. Ferraris, P., Lee, J., Lifschitz, V.: A new perspective on stable models. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007). (2007) 372–379
14. Pearce, D., Valverde, A.: Quantified equilibrium logic and foundations for answer set programs. In: ICLP'08. Volume 5366 of Lecture Notes in Computer Science., Springer (2008) 546–560

15. Doherty, P., Lukaszewicz, W., Szalas, A.: Computing circumscription revisited: A reduction algorithm. J. Autom. Reasoning **18**(3) (1997) 297–336
16. McCarthy, J.: Circumscription: A form of non-monotonic reasoning. Artificial Intelligence **13** (1980) 27–39
17. Heyting, A.: Die formalen Regeln der intuitionistischen Logik. Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse (1930) 42–56
18. Lifschitz, V.: Circumscription. Handbook of Logic in AI and Logic Programming **3** (1994) 298–352
19. Ackermann, W.: Untersuchungen über das eliminationsproblem der mathematischen logik. Mathematische Annalen **110**(1) (1935) 390–413
20. Gabbay, D., Ohlbach, H.J.: Quantifier elimination in second-order predicate logic (1992)

Appendix

(FOR REVIEWING PURPOSES ONLY)

Proof of Theorem 1. We proceed by structural induction.

- If $\varphi = \bot$ then $[\varphi]_i = \bot$ and the result is straightforward.
- If $\varphi = p$ is an atom, then $[p]_i = p(i)$ and we get the chain of equivalent conditions: $\mathbf{M}, i \models p \Leftrightarrow p \in H_i \Leftrightarrow p(i) \in \mathcal{H} \Leftrightarrow \mathcal{M} \models p(i)$.
- If $\varphi = \alpha \wedge \beta$ we get:
  $\quad \mathbf{M}, i \models \alpha \wedge \beta$
  $\Leftrightarrow \mathbf{M}, i \models \alpha$ and $\mathbf{M}, i \models \alpha$
  $\Leftrightarrow \mathcal{M} \models [\alpha]_i$ and $\mathcal{M} \models [\beta]_i$ by induction on α, β
  $\Leftrightarrow \mathcal{M} \models [\alpha]_i \wedge [\beta]_i$
  $\Leftrightarrow \mathcal{M} \models [\alpha \wedge \beta]_i$
- The proof for $\varphi = \alpha \vee \beta$ is analogous to the one for $\alpha \wedge \beta$.
- If $\varphi = \alpha \rightarrow \beta$ we get:
  $\quad \mathbf{M}, i \models \alpha \rightarrow \beta$
  $\Leftrightarrow$ for any $w \in \{\mathbf{H}, \mathbf{T}\}$, $\langle w, \mathbf{T} \rangle, i \models \alpha$ and $\langle w, \mathbf{T} \rangle, i \models \alpha$
  Now, since the THT-interpretation $\langle \mathbf{T}, \mathbf{T} \rangle$ also corresponds to the MHT($<$)-interpretation $\langle \mathcal{T}, \mathcal{T} \rangle$ we can apply induction on subformulas, so that we continue with the equivalent conditions:
  $\Leftrightarrow$ for any $w \in \{\mathcal{H}, \mathcal{T}\}$, $\langle w, \mathcal{T} \rangle \models [\alpha]_i$ and $\langle w, \mathcal{T} \rangle \models [\alpha]_i$
  $\Leftrightarrow \langle \mathcal{H}, \mathcal{T} \rangle \models [\alpha \rightarrow \beta]_i$.
- If $\varphi = \bigcirc\alpha$ we get the equivalent conditions:
  $\quad \mathbf{M}, i \models \bigcirc\alpha$
  $\Leftrightarrow \mathbf{M}, i+1 \models \alpha$
  $\Leftrightarrow \mathcal{M} \models [\alpha]_{i+1}$ by induction
  $\Leftrightarrow \mathcal{M} \models [\bigcirc\alpha]_i$
- If $\varphi = \alpha \, \mathcal{U} \, \beta$ we get the equivalent conditions:
  $\quad \mathbf{M}, i \models \alpha \, \mathcal{U} \, \beta$
  $\Leftrightarrow$ There is some $k \geq i$ s.t. that $\mathbf{M}, k \models \beta$ and for $j \in \{i, \ldots, k-1\}$, $\mathbf{M}, j \models \alpha$
  $\Leftrightarrow$ There is some $k \geq i$ s.t. that $\mathcal{M} \models [\beta]_k$ and for $j \in \{i, \ldots, k-1\}$, $\mathcal{M} \models [\alpha]_j$
  $\Leftrightarrow \mathcal{M} \models \exists k (i \leq k \wedge [\beta]_k \wedge \forall j (i \leq j < k \rightarrow [\alpha]_j))$
  $\Leftrightarrow \mathcal{M} \models [\alpha \, \mathcal{U} \, \beta]_i$
- The proof for $\varphi = \alpha \, \mathcal{R} \, \beta$ is analogous to the one for $\alpha \, \mathcal{U} \, \beta$.

$\blacksquare$