Parametrized equilibrium logic

Ricardo Gonçalves and José Júlio Alferes*

CENTRIA, Universidade Nova de Lisboa, Portugal

Abstract. Equilibrium logic provides a logical foundation for the stable model approach to the semantics of logic programs. Recently, parametrized logic programming was introduced with the aim of presenting the syntax and natural semantics for parametrized logic programs, which are very expressive logic programs, in the sense that complex formulas are allowed to appear in the body and head of rules. Stable model semantics was defined for such parametrized logic programs. The main aim of this paper is to introduce a parametrized version of equilibrium logic and, moreover, to prove that the usual relation with stable model semantics generalizes to the parametrized setting. We also prove that the non-parametrized case can be obtained as a natural choice of the parameter logic. As examples, we obtain general default logic as a particular case and we also show that our approach can be used to characterize and to study strong equivalence of temporal logic programs.

1 Introduction

Equilibrium logic [10], and its monotonic base – the logic of Here-and-There (HT) [7] – provide a logical foundation for the stable models semantics of logic programs [5], in which several important properties of logic programs under the stable model semantics can be studied. In particular, it can be used to check strong equivalence of programs (i.e. that two programs will remain equivalent independently of whatever else we add to both of them), by checking logical equivalence in Here-and-There.

Equilibrium logic also allows for the extension of the stable model semantics to a language allowing arbitrary sets of formulas [4]. In this extension of stable models to general theories, the (disjunctive) logic programming rule symbol "$\leftarrow$" (resp. ",", ";") is equated with implication "$\Rightarrow$" (resp. "$\wedge$", "$\vee$") in equilibrium logic. Moreover, logic programming's default negation is equated with the negation $\neg$ of equilibrium logic. Of course, by doing so the classical connectives of conjunction, disjunction, implication and negation are lost in such general theories. This fact was recently noticed in [15, 16] where equilibrium logic is extended for set of formulas, called general default logic, that besides the equilibrium logic connectives also allows for having classical propositional connectives, e.g. allowing to differentiate logic programming disjunction $A; B$, which indicates that

* The first author was supported by FCT under the postdoctoral grant SFRH/BPD/47245/2008.

either A is known to be true or B is known to be true, from the classical disjunction $A \vee B$ which does not necessarily requires one of the propositions to be known (see [15, 16] for details). This generalisation of equilibrium logic is proven in [15] to be more general than Reiter's default logic [13], and able to express rule constraints, generalised closed world assumption and conditional defaults.

Recently, parametrized logic programming was introduced [6], having in common with [15, 16] the motivation of providing a meaning to theories combining both logic programming connectives with other logic connectives, and allowing complex formulas using these connectives to appear in the head and body of a rule. However, the roots of our approach are quite different, and come from the area of combination of logics [3]. There, the main idea is to fix a parameter logic $\mathcal{L}$ and build up logic programs by replacing atomic sentences with formulas of $\mathcal{L}$. A normal parametrized logic program has the same structure as a normal logic program. The only difference is the fact that atomic symbols are replaced by formulas of a parameter logic. By different choices of parameter logic, different languages and semantics are obtained. As examples, it is shown in [6] how to obtain the answer-set semantics, a paraconsistent version of it, and also the semantics of MKNF hybrid knowledge bases [8].

The main aim of this paper is to introduce a parametrized version of equilibrium logic (Section 3), to prove that the usual relation with stable model semantics generalizes to the parametrized setting, that this parametrized version of equilibrium can be used to prove strong equivalence of parametrized programs, and that the non-parametrized equilibrium logic of [10] can be obtained as a natural choice of the parameter logic. Moreover, we show (Section 3.1) that we obtain general default logic of [15, 16] by choosing as parameter classical propositional logic, this way showing that our approach is more general.

As an example of the usefulness of this extra generality, we explore (Section 4) fixing as parameter linear temporal logic (LTL) [12]. In [1], a temporal extension of Here-and-There, denoted by THT, was introduced to reason about the strong equivalence of temporal logic programs, in order to apply it to transition systems. Although the language of THT is richer than in our parametrized approach, we show in Section 4 that we can capture THT as a particular case and, moreover, we show that there are advantages in adopting a restricted language which does not mix the HT level with the LTL level. One of the advantages is not overloading the LTL classical connectives with the connectives of Here-and-There. Contrary to our approach, in THT (as pointed out in [1]) $\Box$ (always) cannot be defined using $\Diamond$ (eventually) and $\mathcal{R}$ (release) cannot be defined using $\mathcal{U}$ (until). Clearly this is because classical negation and classical implication are not available in THT. Contrarily, we can naturally deal with the unrestricted version of temporal logic programs, i.e, temporal logic programs whose rules can be build using all LTL formulas, thus also including classical negation and classical implication. Moreover, logical equivalence in HT_{LTL} is a necessary and sufficient condition for strong equivalence of normal temporal logic programs whereas logical equivalence in THT is only known to be a sufficient condition for strong equivalence.

2 Parametrized logic programming

In this section we review the main definitions and results of parametrized logic programming [6], necessary for the remainder of the paper. We start by introducing the notion of (monotonic) logic.

Definition 1. *A (monotonic) logic is a pair $\mathcal{L} = \langle L, \vdash_{\mathcal{L}} \rangle$ where L is a set of formulas and $\vdash_{\mathcal{L}}$ is a Tarskian consequence relation [14] over L, i.e. satisfying the following conditions, for every $T \cup \Phi \cup \{\varphi\} \subseteq L$,* **Reflexivity:** *if $\varphi \in T$ then $T \vdash_{\mathcal{L}} \varphi$;* **Cut:** *if $T \vdash_{\mathcal{L}} \varphi$ for all $\varphi \in \Phi$, and $\Phi \vdash_{\mathcal{L}} \psi$ then $T \vdash_{\mathcal{L}} \psi$;* **Weakening:** *if $T \vdash_{\mathcal{L}} \varphi$ and $T \subseteq \Phi$ then $\Phi \vdash_{\mathcal{L}} \varphi$.*

When clear from the context we write $\vdash$ instead of $\vdash_{\mathcal{L}}$. Let $Th(\mathcal{L})$ be the set of *theories of* $\mathcal{L}$, i.e. the set of subsets of L closed under the relation $\vdash_{\mathcal{L}}$. It is well-known that, for every (monotonic) logic $\mathcal{L}$, the tuple $\langle Th(\mathcal{L}), \subseteq \rangle$ is a complete lattice with smallest element the set $Theo = \emptyset^{\vdash}$ of theorems of $\mathcal{L}$ and the greatest element the set L of all formulas of $\mathcal{L}$. Given a subset A of L we denote by $A^{\vdash}$ the smallest theory that contains A. $A^{\vdash}$ is also called the theory generated by A.

In what follows we consider a fixed (monotonic) logic $\mathcal{L} = \langle L, \vdash_{\mathcal{L}} \rangle$ and call it the *parameter logic*. The formulas of $\mathcal{L}$ are dubbed *(parametrized) atoms* and a *(parametrized) literal* is either a parametrized atom φ or its negation *not* φ, where as usual *not* denotes negation as failure. We dub *default literal* those of the form *not* φ.

Definition 2. *A normal $\mathcal{L}$-parametrized logic program is a set of rules of the form $\varphi \leftarrow \psi_1, \ldots, \psi_n, not\ \varphi_1, \ldots, not\ \varphi_m$ where $\varphi, \psi_1, \ldots, \psi_n, \varphi_1, \ldots, \varphi_m \in L$.*

A definite $\mathcal{L}$-parametrized logic program is a set of rules without negations as failure, i.e. of the form $\varphi \leftarrow \psi_1, \ldots, \psi_n$ where $\varphi, \psi_1, \ldots, \psi_n \in L$.

Definition 3. *A (parametrized) interpretation is an $\mathcal{L}$ theory.*

Any interpretation I can be equivalently defined as a function $I : L \to \{0, 1\}$ in the usual way. Given an interpretation I we can extend it to literals by setting $I(not\ \varphi) = 1 - I(\varphi)$ for every $\varphi \in L$.

Definition 4. *A rule $\varphi \leftarrow \psi_1, \ldots, \psi_n, not\ \varphi_1, \ldots, not\ \varphi_m$ is satisfied by an interpretation I if $Min\{I(\psi_1), \ldots, I(\psi_n), I(not\ \varphi_1), \ldots, I(not\ \varphi_m)\} \leq I(\varphi)$.*

A *model* of a program P is an interpretation that satisfies every rule of P. Denote by $Mod_{\mathcal{L}}(P)$ the set of models of P. The stable model semantics of a normal $\mathcal{L}$-parametrized logic programs is defined using a Gelfond-Lifschitz like operator.

Definition 5. *Let P be a normal $\mathcal{L}$-parametrized logic program and I an interpretation. The GL-transformation of P modulo I is the program $\frac{P}{I}$ obtained from P by performing the following operations:*

- *remove from P all rules containing a default literal not φ such that $I \vdash_{\mathcal{L}} \varphi$;*

– *remove from the remaining rules all default literals.*

Since $\frac{P}{I}$ *is a definite* $\mathcal{L}$*-parametrized program, it has an unique least model* J*. We define* $\Gamma(I) = J$*.*

Definition 6. *An interpretation* I *of a* $\mathcal{L}$*-parametrized logic program* P *is a stable model of* P *iff* $\Gamma(I) = I$*. A formula* φ *is true under the stable model semantics iff it belongs to all stable models of* P*.*

We now introduce the important notion of strongly equivalent programs.

Definition 7. *Let* P_1 *and* P_2 *be two normal* $\mathcal{L}$*-parametrized logic programs. We say that* P_1 *and* P_2 *are strongly equivalent if for every normal* $\mathcal{L}$*-parametrized logic program* P*, the programs* $P_1 \cup P$ *and* $P_2 \cup P$ *have the same stable models.*

3 Parametrized equilibrium logic

In this section we present the main proposal of this paper: a parametrized version of equilibrium logic and of its monotonic base, the logic of Here-and-There (HT) [7], generalising parametrized logic programming.

Given a parameter logic $\mathcal{L} = \langle L, \vdash \rangle$, we dub $HT_{\mathcal{L}}$ the $\mathcal{L}$-parametrized logic of Here-and-There. The language of $HT_{\mathcal{L}}$ is build constructively from the formulas of $\mathcal{L}$ using the connectives $\bot$, $\wedge$, $\vee$ and $\rightarrow$. Negation $\neg$ is introduced as an abbreviation $\neg\delta := (\delta \rightarrow \bot)$. Note that the formulas of the parameter logic $\mathcal{L}$ act as atoms of the $HT_{\mathcal{L}}$ language.

The semantics of $HT_{\mathcal{L}}$ is a generalization of the intuitionistic Kripke semantics of HT. A frame for $HT_{\mathcal{L}}$ is a tuple $\langle W, \leq \rangle$ where W is a set of exactly two worlds, say h (here) and t (there) with $h \leq t$. A $HT_{\mathcal{L}}$ interpretation is a frame together with an assignment i that associates to each world a theory of $\mathcal{L}$, such that $i(h) \subseteq i(t)$. Note the key idea of substituting, in the original definition of HT interpretations, sets of atoms by theories of the parameter logic, i.e., by sets of $\mathcal{L}$ formulas closed under $\vdash_{\mathcal{L}}$. An interpretation is said to be *total* if $i(h) = i(t)$. It is convenient to see a $HT_{\mathcal{L}}$ interpretation as an ordered pair $\langle T^h, T^t \rangle$ such that $T^h = i(h)$ and $T^t = i(t)$ where i is the interpretation's assignment. We define the satisfaction relation between a $HT_{\mathcal{L}}$ interpretation $\langle T^h, T^t \rangle$ at a particular world w and a $HT_{\mathcal{L}}$ formula δ recursively, as follows:

i) for $\varphi \in L$ we have that $\langle T^h, T^t \rangle, w \Vdash \varphi$ if $T^w \vdash_{\mathcal{L}} \varphi$;
ii) $\langle T^h, T^t \rangle, w \nVdash \bot$;
iii) $\langle T^h, T^t \rangle, w \Vdash (\delta_1 \vee \delta_2)$ if $\langle T^h, T^t \rangle, w \Vdash \delta_1$ or $\langle T^h, T^t \rangle, w \Vdash \delta_2$;
iv) $\langle T^h, T^t \rangle, w \Vdash (\delta_1 \wedge \delta_2)$ if $\langle T^h, T^t \rangle, w \Vdash \delta_1$ and $\langle T^h, T^t \rangle, w \Vdash \delta_2$;
v) $\langle T^h, T^t \rangle, w \Vdash (\delta_1 \rightarrow \delta_2)$ if for every w' such that $w \leq w'$ we have that $\langle T^h, T^t \rangle, w' \nVdash \delta_1$ or $\langle T^h, T^t \rangle, w' \Vdash \delta_2$.

Note that for $\varphi \in L$ and since T^w is a $\mathcal{L}$ theory, the condition $T^w \vdash_{\mathcal{L}} \varphi$ is equivalent to $\varphi \in T^w$. We say that an interpretation $\mathcal{I}$ is a model of a $HT_{\mathcal{L}}$ formula δ if $\mathcal{I}, w \Vdash \delta$ for every $w \in \{h, t\}$. As in the usual case, this is equivalent to the single fact $\mathcal{I}, h \Vdash \delta$. A formula δ is said to be a consequence of a set of formulas Φ, denoted by $\Phi \vdash_{HT_{\mathcal{L}}} \delta$, if for every interpretation $\mathcal{I}$ and every world w we have that $\mathcal{I}, w \Vdash \delta$ whenever $\mathcal{I}, w \Vdash \delta'$ for every $\delta' \in \Phi$.

Definition 8. *An equilibrium model of a set Φ of $HT_{\mathcal{L}}$ formulas is a total $HT_{\mathcal{L}}$ interpretation $\langle T, T\rangle$ such that*

1. *$\langle T, T\rangle$ is a model of Φ;*
2. *for every $\mathcal{L}$ theory $T' \subset T$ we have that $\langle T', T\rangle$ is not a model of Φ.*

With this notion of model, equilibrium entailment is defined as follows:

Definition 9. *The equilibrium entailment, $\models_E$, over $HT_{\mathcal{L}}$ formulas is defined for every set $\Phi \cup \{\delta\}$ of $HT_{\mathcal{L}}$ formulas as follows:*

- *if Φ is non-empty and has equilibrium models then $\Phi \models_E \delta$ if every equilibrium model of Φ is a $HT_{\mathcal{L}}$ model of δ;*
- *if Φ is empty or does not have equilibrium models then $\Phi \models_E \delta$ if $\Phi \vdash_{HT_{\mathcal{L}}} \delta$.*

Clearly, the traditional HT logic is a particular case of our parametrized approach. This can be seen by using a natural choice $\mathcal{L} = \langle L, \vdash_{\mathcal{L}}\rangle$ of the parameter logic, where the language L of $\mathcal{L}$ is the set At of atoms of HT. Then, the only reasonable consequence definable over L is the trivial one, i.e., for every $X \cup \{p\} \subseteq At$ we have $X \vdash_{\mathcal{L}} p$ iff $p \in X$. In this case the $\mathcal{L}$ theories ($HT_{\mathcal{L}}$ interpretations) are nothing but sets of atoms (HT interpretations). So, HT coincides with $HT_{\mathcal{L}}$. This implies that equilibrium logic can also be captured as a particular case of our approach.

We now proceed by proving that parametrized equilibrium logic coincides with [6] in the specific case of logic programs (Lemma 3), and that $HT_{\mathcal{L}}$ can be used to prove strong equivalence of parametrized logic programs (Theorem 1). For that, we identify a rule $\varphi \leftarrow \psi_1, \ldots, \psi_n, not\ \varphi_1, \ldots, not\ \varphi_m$ of a $\mathcal{L}$-parametrized program with the $HT_{\mathcal{L}}$-formula $(\psi_1 \wedge \ldots \wedge \psi_n \wedge \neg\varphi_1 \wedge \ldots \wedge \neg\varphi_m) \rightarrow \varphi$, and a program P with the set of $HT_{\mathcal{L}}$-formulas that correspond to its rules.

Lemma 1. *An interpretation $\langle T^h, T^t\rangle$ is a model of a definite $\mathcal{L}$-parametrized program P iff both T^h and T^t are models of P.*

Proof. Let P be a definite $\mathcal{L}$-parametrized program. A $HT_{\mathcal{L}}$ interpretation $\mathcal{I}$ is a model of P iff for every rule $\varphi \leftarrow \psi_1, \ldots, \psi_n \in P$ and every $w \in \{h, t\}$ we have $\mathcal{I}, w \Vdash \varphi$ whenever $\mathcal{I}, w \Vdash \psi_i$ for every $i \in \{1, \ldots, n\}$. This amounts to say that T^h and T^t both satisfy each rule of P, i.e., T^h and T^t are both models of P. ■

Lemma 2. *An interpretation $\langle T^h, T^t\rangle$ is a model of a $\mathcal{L}$-parametrized program P iff it is a model of $\frac{P}{T^t}$.*

Proof. Recall that $\frac{P}{T^t}$ is obtained by dropping every rule r of P such that $not\ \psi \in r$ and $\psi \in T^t$ and by eliminating all default atoms from the remaining rules.

Suppose first that $\langle T^h, T^t\rangle$ is a model of P. We aim to prove that it is also a model of $\frac{P}{T^t}$. A rule $\varphi \leftarrow \psi_1, \ldots, \psi_n \in \frac{P}{T^t}$ is obtained from the rule $\varphi \leftarrow \psi_1, \ldots, \psi_n, not\ \varphi_1, \ldots, not\ \varphi_m \in P$ where $\varphi_1, \ldots, \varphi_m \notin T^t$. Therefore, we have that $\langle T^h, T^t\rangle \Vdash \neg\varphi_i$ for every $i \in \{1, \ldots, m\}$, and we can conclude that $\langle T^h, T^t\rangle \Vdash (\psi_1 \wedge \ldots \wedge \psi_n) \rightarrow \varphi$ iff $\langle T^h, T^t\rangle \Vdash (\psi_1 \wedge \ldots \wedge \psi_n \wedge \neg\varphi_1 \wedge \ldots \wedge \neg\varphi_m) \rightarrow \varphi$. The later holds because $\langle T^h, T^t\rangle$ is assumed to be a model of P.

Now assume that $\langle T^h, T^t\rangle$ is a model of $\frac{P}{T^t}$. We aim to prove that it is also a model of P. Let $r = \varphi \leftarrow \psi_1, \ldots, \psi_n, not\ \varphi_1, \ldots, not\ \varphi_m \in P$. Then we have two cases: *Case 1:* there exists some $i \in \{1, \ldots, m\}$ such that $\varphi_i \in T^t$. Then, $\langle T^h, T^t\rangle \nVdash \neg\varphi_i$ and trivially we have that $\langle T^h, T^t\rangle$ satisfies r.
Case 2: For every $i \in \{1, \ldots, m\}$ we have that $\varphi_i \notin T^t$. Then $\langle T^h, T^t\rangle \Vdash \neg\varphi_i$ for every $i \in \{1, \ldots, m\}$. Therefore, $\langle T^h, T^t\rangle \Vdash (\psi_1 \wedge \ldots \wedge \psi_n \wedge \neg\varphi_1 \wedge \ldots \wedge \neg\varphi_m) \rightarrow \varphi$ iff $\langle T^h, T^t\rangle \Vdash (\psi_1 \wedge \ldots \wedge \psi_n) \rightarrow \varphi$. The later holds because $\langle T^h, T^t\rangle$ is assumed to be a model of $\frac{P}{T^t}$ and $\varphi \leftarrow \psi_1, \ldots, \psi_n \in \frac{P}{T^t}$. So, we can conclude that $\langle T^h, T^t\rangle$ satisfies every rule of P and it is, therefore, a model of P. ■

Lemma 3. *For any $\mathcal{L}$-parametrized program P, an $HT_\mathcal{L}$ interpretation $\langle T, T\rangle$ is an equilibrium model of P iff T is stable model of P.*
Proof. T is a stable model of P iff T is a stable model of $\frac{P}{T}$ iff T is a model of $\frac{P}{T}$ and for every theory $T' \subset T$, T' is a not model of $\frac{P}{T}$. Using Lemma 1, this can be equivalently restated as $\langle T, T\rangle$ is a model of $\frac{P}{T}$ and for every theory $T' \subset T$ $\langle T', T\rangle$ is a not model of $\frac{P}{T}$. By Lemma 2 this is equivalent to the conditions: $\langle T, T\rangle$ is a model of P and for every theory $T' \subset T$, $\langle T', T\rangle$ is not a model of P. By definition, this means that $\langle T, T\rangle$ is an equilibrium model of P. ■

Lemma 4. *For any $\mathcal{L}$-parametrized programs P_1 and P_2 the following conditions are equivalent: a) for every program P the programs $P_1 \cup P$ and $P_2 \cup P$ have the same equilibrium models; b) P_1 and P_2 are equivalent in $HT_\mathcal{L}$.*
Proof. First suppose that P_1 and P_2 are equivalent in $HT_\mathcal{L}$. Then, it is clear that $P_1 \cup P$ and $P_2 \cup P$ are also equivalent and, in particular, have the same equilibrium models. We now prove the reverse direction by contraposition. Without loss of generality, suppose that P_1 has a model $\langle T^h, T^t\rangle$ which is not a model of P_2. We show how to find a program P such that $\langle T^t, T^t\rangle$ is an equilibrium model of one of the programs $P_1 \cup P$ or $P_2 \cup P$ but not an equilibrium model of the other. We have two cases: *Case 1:* $\langle T^t, T^t\rangle$ is not a model of P_2. It is easy to see that it is a model of P_1. We then take $P = T^t$. Clearly $\langle T^t, T^t\rangle$ is a model of $P_1 \cup T^t$. Since for every $T \subset T^t$ we have that $\langle T, T^t\rangle$ is not a model of $P_1 \cup T^t$, we can conclude that $\langle T^t, T^t\rangle$ is an equilibrium model of $P_1 \cup T^t$. On the other hand $\langle T^t, T^t\rangle$ is not even a model of $P_2 \cup T^t$ since it is not a model of P_2.
Case 2: $\langle T^t, T^t\rangle$ is a model of P_2. Take $P = T^h \cup \{\varphi \rightarrow \psi : \varphi, \psi \in T^t \setminus T^h, \varphi \neq \psi\}$. Clearly $\langle T^t, T^t\rangle$ is a model of P and it is also a model of $P_2 \cup P$. Let us now prove that it is in fact a equilibrium model. Suppose that there exists $T \subset T^t$ such that $\langle T, T^t\rangle$ is a model of $P_2 \cup P$. By definition of P, we have $T^h \subseteq T$, but $T \neq T^h$ because $\langle T^h, T^t\rangle$ is not a model of P_2. Therefore, $T^h \subset T \subset T^t$. If we take $\varphi \in T \setminus T^h$ and $\psi \in T^t \setminus T$ we have that $\langle T, T^t\rangle$ does not satisfy $\varphi \rightarrow \psi \in P$. But this contradicts the fact that $\langle T, T^t\rangle$ is a model of $P_2 \cup P$. It remains to be checked that $\langle T^t, T^t\rangle$ is not an equilibrium model of $P_1 \cup P$. This follows from the straightforward fact that $\langle T^h, T^t\rangle$ is a model of $P_1 \cup P$, along with the fact that $T^h \neq T^t$, because $\langle T^t, T^t\rangle$ is a model of P_2 but $\langle T^h, T^t\rangle$ is not. ■

The following is the main theorem of this section and it follows straightforwardly from Lemma 4 and Lemma 3.

Theorem 1. *Let P_1 and P_2 be two $\mathcal{L}$-parametrized logic programs. The following conditions are equivalent: P_1 and P_2 are strongly equivalent; P_1 and P_2 are equivalent in $HT_{\mathcal{L}}$.*

One application of the notion of strong equivalence between logic program is the simplification of logic programs [9]. A logic program P is a simplification of a program P' if P is simpler than P' and P is strongly equivalent to P'. Here, we do not deal with the general problem of the simplification of logic programs. Still, in the context of parametrized logic program, a particular simplification holds. In fact, it is easy to prove that by simplifying the atoms of a parametrized logic program P (using logical equivalence in the parameter logic) we obtain a parametrized logic program which is a simplification of P.

3.1 Comparison with General default logic

We now compare our approach with general default logic (GDL) of [15, 16]. We show that GDL can be obtained as a particular case of our approach using classical propositional logic (CPL) as parameter logic.

First we introduce the language of GDL. Let $CPL = \langle L_{CPL}, \vdash_{CPL} \rangle$ denotes classical propositional logic, where L_{CPL} is built from a set $\mathcal{P}$ of propositional symbols using the usual classical connectives $(\Rightarrow, \sqcap, \sqcup, \mathbf{f})$. Classical negation is defined as $\sim \varphi := (\varphi \Rightarrow \mathbf{f})$. The consequence relation $\vdash_{CPL}$ is the usual classical consequence. The language of GDL, here denoted by L_{GDL}, is obtained constructively from CPL formulas using the HT connectives $\rightarrow, \vee$ and $\wedge$. Note that L_{GDL} does not contain $\bot$. In [15] the authors define a derived negation, here denoted by $\neg'$, using $\mathbf{f}$, i.e, $\neg'\delta := (\delta \rightarrow \mathbf{f})$. Nevertheless, $\neg'$ does not coincide with HT negation (recall that HT negation is defined as $\neg\delta := (\delta \rightarrow \bot)$). It is, nevertheless, straightforward to extend the semantics given in [15, 16] in order to include $\bot$ (and, therefore, $\neg$) in the language of GDL. So, it is clear that L_{GDL} coincides with the language of HT_{CPL} (as defined in Section 3).

Two different semantics have been defined for the language of GDL. The semantics defined in [15] is along the lines of the stable model semantics, using a proper notion of program reduct, whereas in [16] the semantics is defined along the lines of the equilibrium logic approach.

Being both based in the equilibrium logic approach, the semantics defined in [16] and HT_{CPL} can be directly compared. In fact, by construction, it is immediate that they coincide, that is, the HT semantics defined in [16] is obtained by taking, in our approach, CPL as parameter logic.

In [16] the authors draw a strong connection between their HT semantics and the stable models like semantics of [15]. Concretely, they prove that if T is a CPL theory and Φ a set of GDL formulas then, $\langle T, T \rangle$ is an equilibrium model of Φ iff T is an extension (generalization of a stable model) of Φ. Since HT_{CPL} coincides with the HT semantics defined in [16], this strong connection also holds for HT_{CPL}.

4 Example: Temporal Here-and-There logic

As we have proven that our approach is more general than general defaults, in that we can use other (parameter) logics, it is worth showing that this extra generality is useful. We do that in this section by experimenting with linear temporal logic LTL [12] as parameter logic. We compare the obtained logic with the Temporal Here-and-There logic THT of [1], a generalization of temporal logic programs that can be used for expressing knowledge in the domain of actions.

4.1 Logics of LTL and THT

In order to make the paper complete we introduce LTL [12] and THT [1]. We start with LTL. The language of LTL is build from a set of propositional symbols $\mathcal{P}$ using the usual classical connectives $\mathbf{t}, \sim, \sqcap, \sqcup, \Rightarrow$, the unary temporal operator $\bigcirc$ (next) and the binary temporal operator $\mathcal{U}$ (until). The other usual temporal operators can be defined by abbreviation: $\Diamond\varphi := \mathbf{t}\mathcal{U}\varphi$ (eventually), $\Box\varphi :=\sim\Diamond\sim\varphi$ (always), $\varphi_1\mathcal{W}\varphi_2 := (\varphi_1\mathcal{U}\varphi_2) \sqcup (\Box\varphi_1)$ (weak until) and $\varphi_1\mathcal{R}\varphi_2 :=\sim((\sim\varphi_1)\mathcal{U}(\sim\varphi_2))$ (release).

A LTL interpretation is a sequence $m = (m_i)_{i\in\mathbb{N}}$ where $m_i \subseteq \mathcal{P}$ for each $i \in \mathbb{N}$. The satisfaction of a formula by a LTL interpretation $m = (m_i)_{i\in\mathbb{N}}$ at a point i is defined inductively as follows:

- $m, i \Vdash p$ if $p \in m_i$, for $p \in \mathcal{P}$;
- $m, i \Vdash\sim\varphi$ if $m, i \nVdash \varphi$;
- $m, i \Vdash \varphi_1 \sqcap \varphi_2$ if $m, i \Vdash \varphi_1$ and $m, i \Vdash \varphi_2$;
- $m, i \Vdash \varphi_1 \sqcup \varphi_2$ if $m, i \Vdash \varphi_1$ or $m, i \Vdash \varphi_2$;
- $m, i \Vdash \varphi_1 \Rightarrow \varphi_2$ if $m, i \nVdash \varphi_1$ or $m, i \Vdash \varphi_2$;
- $m, i \Vdash \bigcirc\varphi$ if $m, i + 1 \Vdash \varphi$;
- $m, i \Vdash \varphi_1\mathcal{U}\varphi_2$ if $m, j \Vdash \varphi_2$ for some $j \geq i$ and $m, k \Vdash \varphi_1$ for every $i \leq k < j$.

We say that a LTL interpretation m is a model of a LTL formula φ if $m, 0 \Vdash \varphi$. This is the so-called anchored version of LTL. Given a set Φ of LTL formulas we denote by $Mod(\Phi)$ the set of LTL models of all formulas of Φ. A LTL formula φ is valid if $m \Vdash \varphi$ for every LTL interpretation m. The consequence relation $\vdash_{LTL}$ is defined as usual, i.e., $\Phi \vdash_{LTL} \varphi$ if for every interpretation m we have that $m \Vdash \varphi$ whenever $m \Vdash \psi$ for every $\psi \in \Phi$.

We now introduce (THT) of [1]. The language of THT is build from the set $\mathcal{P}$ of propositional symbols using interchangeably HT connectives ($\bot, \rightarrow, \vee, \wedge$) and LTL temporal operators ($\bigcirc, \Box, \Diamond, \mathcal{U}, \mathcal{W}, \mathcal{R}$). Note that in THT the operators $\bigcirc$ and $\mathcal{U}$ are not considered the only primitive temporal operators. This is due to the fact that in THT some usual inter definability relations between temporal operators do not hold, that is, some operators can not be defined defined as abbreviations. In turn, this is intimately related with the fact that the language of THT does not contain classical negation and implication. Below we will come back to this issue with more detail.

A THT interpretation is a pair $M = \langle m^h, m^t\rangle$ where m^h, m^t are LTL interpretations such that $m^h_i \subseteq m^t_i$ for every $i \in \mathbb{N}$. Being a combination of LTL with

HT, THT interpretations have, is some sense, two dimensions. In the HT dimension we have the two worlds h (here) and t (there) and in the LTL dimension we have the time instants $i \in \mathbb{N}$. Therefore, the satisfaction of a THT formula δ by a THT interpretation $M = \langle m^h, m^t \rangle$ is defined at a world $w \in \{h, t\}$ and at a time instant $i \in \mathbb{N}$, by induction on the structure of δ:

- $\langle m^h, m^t \rangle, w, i \nVdash \bot$;
- $\langle m^h, m^t \rangle, w, i \Vdash p$ if $p \in m_i^w$, for $p \in \mathcal{P}$;
- $\langle m^h, m^t \rangle, w, i \Vdash \bigcirc\varphi$ if $\langle m^h, m^t \rangle, w, i+1 \Vdash \varphi$;
- $\langle m^h, m^t \rangle, w, i \Vdash \Box\varphi$ if $\langle m^h, m^t \rangle, w, j \Vdash \varphi$ for every $j \geq i$;
- $\langle m^h, m^t \rangle, w, i \Vdash \varphi_1 \mathcal{U} \varphi_2$ if $\langle m^h, m^t \rangle, w, j \Vdash \varphi_2$ for some $j \geq i$ and $\langle m^h, m^t \rangle, w, k \Vdash \varphi_1$ for every $i \leq k < j$;
- $\langle m^h, m^t \rangle, w, i \Vdash \varphi_1 \mathcal{R} \varphi_2$ if for all $j \geq i$ either $\langle m^h, m^t \rangle, w, j \Vdash \varphi_2$ or there exists k, $i \leq k < j$ such that $\langle m^h, m^t \rangle, w, k \Vdash \varphi_1$;
- $\langle m^h, m^t \rangle, w, i \Vdash (\delta_1 \vee \delta_2)$ if $\langle m^h, m^t \rangle, w, i \Vdash \delta_1$ or $\langle m^h, m^t \rangle, w, i \Vdash \delta_2$;
- $\langle m^h, m^t \rangle, w, i \Vdash (\delta_1 \wedge \delta_2)$ if $\langle m^h, m^t \rangle, w, i \Vdash \delta_1$ and $\langle m^h, m^t \rangle, w, i \Vdash \delta_2$;
- $\langle m^h, m^t \rangle, w, i \Vdash (\delta_1 \rightarrow \delta_2)$ if for every w' such that $w \leq w'$ we have that $\langle m^h, m^t \rangle, w', i \nVdash \delta_1$ or $\langle m^h, m^t \rangle, w', i \Vdash \delta_2$.

A THT interpretation M is a model of a THT formula φ, denoted by $M \Vdash \varphi$ if $M, h, 0 \Vdash \varphi$. This definition also uses the anchored version of LTL. Given a set Φ of THT formulas we denote by $Mod(\Phi)$ the set of THT interpretations that are models of every formula of Φ. A THT formula δ is valid if every THT interpretation is a model of δ. The consequence relation $\vdash_{THT}$ can be defined as $\Phi \vdash_{THT} \delta$ if for every interpretation M and world $w \in \{h, t\}$ we have that $M, w \Vdash \delta$ whenever $M, w \Vdash \delta'$ for every $\delta' \in \Phi$.

Recall that the language of THT is built using interchangeably HT connectives and LTL temporal operators. In HT_{LTL} (HT parametrized with LTL, as defined in Section 3) this interaction between the HT level and the parameter logic level is not allowed. For example, $\Box(p \rightarrow q)$ is not a HT_{LTL} formula. Therefore, it is clear that THT as a richer language than HT_{LTL}. Still, there are advantages in adopting a restricted language which does not mix the HT level with the LTL level. One immediate advantage is the fact that in HT_{LTL} we do not overload classical negation and implication with HT negation and implication. Contrarily, the language of THT does not contain these two classical connectives. The THT connectives $\rightarrow$ and $\neg$ are intuitionistic. For example, $\neg p \vee p$ is not a valid formula in THT. Moreover, if the language of THT is enriched with classical implication or classical negation then THT is no longer a conservative extension of HT since, for example, the HT valid formula $\neg\varphi \vee \neg\neg\varphi$ is not valid in the enriched THT. What is at the heart of this limitation is the fact that THT interpretations are build up using the semantics of LTL. Let us detail. Recall that a THT interpretation is a pair $M = \langle m^h, m^t \rangle$ where m^h, m^t

are LTL interpretations such that $m_i^h \subseteq m_i^t$ for every $i \in \mathbb{N}$. But the condition $m_i^h \subseteq m_i^t$, for every $i \in \mathbb{N}$, does not guarantee the intuitionistic persistence property for LTL formulas: if a LTL formula is true at h then it is also true at t. Clearly, in general this property fails when the LTL formula contains classical negation or implication. Therefore, the persistency property holds in THT only because classical negation and implication are not part of the language. Contrarily, in our approach we use theories of the parameter logic, thus our construction of HT interpretations is always independent of the available semantics of the parameter logic. This independency of the semantical presentation of the parameter logic is a keystone property of our approach.

In [2] a normal form closely related with logic program rules was obtained for THT formulas. In more detail, in [2] it is proved that every THT formula is THT equivalent to a formula of the following form:

- an atom $p \in \mathcal{P}$;
- $\Box(B_1 \wedge \ldots \wedge B_n \rightarrow (C_1 \vee \ldots \vee C_m))$ where for each B_i and each C_j is a temporal literal, that is, of the form p, $\bigcirc p$ or $\neg p$ with $p \in \mathcal{P}$;
- $\Box(\Box p \rightarrow q)$ or $\Box(p \rightarrow \Diamond q)$ for some $p, q \in \mathcal{P}$.

Therefore, although the language of THT is richer than that of HT_{LTL}, every formula of THT as a normal form which is almost a formula of HT_{LTL}. It is "almost" because of the temporal operator $\Box$ that embraces the formula. Still, we will show that we can use our parametrized approach to reason about this kind of formulas.

4.2 Strong equivalence of temporal logic programs with parametrized *HT*

In the last section we have discussed the differences between the languages of THT and HT_{LTL}. In this section, our aim is to relate the equivalence in these two logics. Of course, in order to relate them we need to restrict to a common language. Recall that the results proved in Section 3 allow us to use HT_{LTL} to characterize and reason about strong equivalence of normal LTL-parametrized logic programs, i. e., with rules of the form $\varphi \leftarrow \varphi_1, \ldots, \varphi_n, not\ \psi_1, \ldots, not\ \psi_m$ where $\varphi, \varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m$ are LTL formulas. Still, we do not restrict our study to normal LTL-parametrized logic programs. We will use HT_{LTL} to reason about a more general kind of programs.

By a normal temporal logic program we mean a set of rules of the form $\Box(\varphi \leftarrow \varphi_1, \ldots, \varphi_n, not\ \psi_1, \ldots, not\ \psi_m)$ where $\varphi, \varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m$ are LTL formulas. We denote by LTL^- the LTL fragment of THT, i.e., LTL restricted to the language built from $\mathcal{P}$ using just the temporal operators $(\bigcirc, \Box, \Diamond, \mathcal{U}, \mathcal{W}, \mathcal{R})$. In order to stay in a common language we focus our attention on the class of normal temporal logic programs whose rules are built just from LTL^- formulas. We will identify a rule $\Box(\varphi \leftarrow \varphi_1, \ldots, \varphi_n, not\ \psi_1, \ldots, not\ \psi_m)$ with the THT formula $\Box((\varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\psi_1 \wedge \ldots \wedge \neg\psi_m) \rightarrow \varphi)$.

The first problem we face is the fact that the THT formula obtained from a normal temporal logic program rule is not a HT_{LTL} formula. Let us now

show how can we solve this problem. Recall that given a rule $r = \Box(\varphi \leftarrow \varphi_1, \ldots, \varphi_n, not\ \psi_1, \ldots, not\ \psi_m)$ of a normal temporal logic program and a THT interpretation M we have that M is a model of r iff $M, h, i \Vdash (\varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\psi_1 \wedge \ldots \wedge \neg\psi_m) \rightarrow \varphi$ for every $i \in \mathbb{N}$. Therefore, the intended meaning of r is that the rule $\varphi \leftarrow \varphi_1, \ldots, \varphi_n, not\ \psi_1, \ldots, not\ \psi_m$ holds at each time instant $i \in \mathbb{N}$. To syntactically express that a LTL formula holds at a particular time instance we consider, for every LTL formula φ and every $i \in \mathbb{N}$, the LTL formula $\varphi^i := \bigcirc^i\varphi$, that is, φ preceded by i occurrences of $\bigcirc$.

Let $r = \Box(\varphi \leftarrow \varphi_1, \ldots, \varphi_n, not\ \psi_1, \ldots, not\ \psi_m)$ be a normal temporal logic program rule and consider the set $r^* = \{\varphi^i \leftarrow \varphi_1^i, \ldots, \varphi_n^i, not\ \psi_1^i, \ldots, not\ \psi_m^i : i \in \mathbb{N}\}$ of HT_{LTL} parametrized rules obtained from r. Given a normal temporal logic program P we consider the HT_{LTL} parametrized normal logic program $P^* = \bigcup_{r \in P} r^*$. Note that even for a finite program P, the resulting program P^* is infinite. This is a natural consequence of the infinite flavor of $\Box$.

We now compare logical equivalence of temporal logic programs in HT_{LTL} and in THT. More precisely, we will prove that two temporal logic programs P_1 and P_2 are equivalent in THT iff P_1^* and P_2^* are equivalent in HT_{LTL}[1].

In order to compare HT_{LTL} and THT, we need to bear in mind the major difference between these two approaches[2]: a HT_{LTL} interpretation is a pair of LTL theories whereas a THT interpretation is a pair of LTL interpretations. Therefore, we start by giving a strong connection between LTL theories and LTL interpretations. Concretely, we define, for every LTL theory T, a LTL interpretation satisfying exactly the formulas in T and, conversely, we define, for every LTL interpretation m, a LTL theory containing exactly the LTL formulas satisfied by m.

Let T be a LTL theory and consider the LTL interpretation $m^T = (m_i^T)_{i \in \mathbb{N}}$ obtained from T where, for every $i \in \mathbb{N}$, $m_i^T = \bigcap_{m \in Mod_{LTL}(T)} m_i$. Conversely, given a LTL interpretation m consider the LTL theory $Th(m) = \{\varphi : m \Vdash \varphi\}$.

We now extend this relation between LTL theories and LTL interpretations to a relation between HT_{LTL} interpretations, which are based on LTL theories, and THT interpretations, which are based on LTL interpretations. First define, for every HT_{LTL} interpretation $M = \langle T^h, T^t \rangle$, the THT interpretation $M^{t2m} = \langle m^{T^h}, m^{T^t} \rangle$. Note that M^{t2m} satisfies the condition in the definition of a THT interpretation, that is, $m^{T^h} \subseteq m^{T^t}$ only because we are restricted to LTL^- formulas. This would not be true in the full LTL language including classical negation and implication.

Now define for every THT interpretation $M = \langle m^h, m^t \rangle$ a HT_{LTL} interpretation $M^{m2t} = \langle Th(m^h), Th(m^t) \rangle$. Again, M^{m2t} satisfies the condition in the definition of a HT_{LTL} interpretation, that is, $Th(m^h) \subseteq Th(m^t)$, only because we are restricted to LTL^- formulas.

---

[1] For lack of space we have omitted the proofs. Nevertheless, the detailed proofs can be found in `http://www.centria.fct.unl.pt/~rgon/articles/HTparametrized.pdf`.

[2] Of course, the first main difference between HT_{LTL} and THT is the language. But now we are restricted to the common language of temporal logic programs.

The following lemma is straightforward from the definition of m^T.

Lemma 5. *Let T be a LTL theory and φ a LTL^- formula.*
Then, for every $i \in \mathbb{N}$ we have that $m^T, i \Vdash \varphi$ iff $m, i \Vdash \varphi$ for every $m \in Mod_{LTL}(T)$.

Lemma 6. *Let T be a LTL theory and φ an LTL^- formula.*
Then, $m^T, i \Vdash \varphi$ iff $\varphi^i \in T$.

Proof. Consider the following sequence of equivalent sentences: $m^T, i \Vdash \varphi$ iff (Lemma 5) $m, i \Vdash \varphi$ for every $m \in Mod_{LTL}(T)$ iff $m, 0 \Vdash \bigcirc^i \varphi$ for every $m \in Mod_{LTL}(T)$ iff $T \vdash \varphi^i$ iff (T is a LTL theory) $\varphi^i \in T$. ■

The next corollary guarantees that M^{t2m} is well-defined, i.e., M^{t2m} satisfies the condition in the definition of a THT interpretation.

Corollary 1. *Let T_1 and T_2 be a LTL theories such that $T_1 \subseteq T_2$. Then for every $i \in \mathbb{N}$ we have that $m_i^{T_1} \subseteq m_i^{T_2}$.*

Proof. The result follows easily from the observation that Lemma 6 implies that, for every LTL theory T and every $p \in P$, we have that $p \in m_i^T$ iff $p^i \in T$. ■

Lemma 7. *Let m be a LTL interpretation. Then $Th(m)$ is a LTL theory.*

Proof. Suppose that $Th(m) \vdash_{LTL} \varphi$. Then, for every LTL interpretation m', we have that $m' \Vdash_{LTL} \varphi$ whenever $m' \in Mod_{LTL}(Th(m))$. Since clearly $m \in Mod_{LTL}(Th(m))$ we can conclude that $m \Vdash_{LTL} \varphi$, i.e., $\varphi \in Th(m)$. ■

The next Lemma guarantees that M^{m2t} is well-defined, i.e., M^{m2t} satisfies the condition in the definition of a HT_{LTL} interpretation.

Lemma 8. *Let $m = (m_i)_{i \in \mathbb{N}}$ and $m' = (m'_i)_{i \in \mathbb{N}}$ be LTL interpretations. If $m_i \subseteq m'_i$ for every $i \in \mathbb{N}$ then $Th(m) \subseteq Th(m')$.*

Proof. We prove, by induction on the structure of a LTL^- formula δ, that $m' \Vdash \delta$ whenever $m \Vdash \delta$. This immediately implies that $Th(m) \subseteq Th(m')$.

Base: $\delta = p$ where $p \in P$. Then, $m \Vdash_{LTL} p$ iff $m, 0 \Vdash p$ iff $p \in m_0$ then $p \in m'_0$ iff $m', 0 \Vdash p$ iff $m' \Vdash p$.

Step: We just consider the case $\delta = (\delta_1 \mathcal{U} \delta_2)$. The other cases can be done in a very similar way. Suppose that $m \Vdash \delta_1 \mathcal{U} \delta_2$. Then, $m, 0 \Vdash_{LTL} \delta_1 \mathcal{U} \delta_2$ and, therefore, there exists $j \geq 0$ such that $m, j \Vdash \delta_2$ and for every $0 \leq k < j$ we have that $m, k \Vdash \delta_1$. By induction hypothesis we have that there exists $j \geq 0$ such that $m', j \Vdash \delta_2$ and for every $0 \leq k < j$ we have that $m', k \Vdash \delta_1$. Then we can conclude that $m', 0 \Vdash \delta_1 \mathcal{U} \delta_2$ and, therefore, $m' \Vdash \delta_1 \mathcal{U} \delta_2$. ■

Theorem 2. *Let $M = \langle T^h, T^t \rangle$ be a HT_{LTL} interpretation. Then, for every $w \in \{h, t\}$, every $i \in \mathbb{N}$ and every rule $r = \varphi \leftarrow \varphi_1, \ldots, \varphi_n, not\ \psi_1, \ldots, not\ \psi_m$ we have that $M, w \Vdash_{HT_{LTL}} r^i$ iff $M^{t2m}, w, i \Vdash_{THT} r$.*

Proof. Let $w \in \{h,t\}$ and $i \in \mathbb{N}$. Then $M,w \Vdash_{HT_{LTL}} r^i$ iff $M,w \Vdash_{HT_{LTL}} (\varphi_1^i \wedge \ldots \wedge \varphi_n^i \wedge \neg\psi_1^i \wedge \ldots \wedge \neg\psi_m^i) \rightarrow \varphi^i$ iff $M,w \Vdash_{HT_{LTL}} \varphi^i$ whenever $M,w \Vdash_{HT_{LTL}} \varphi_j^i$ for every $j \in \{1,\ldots,n\}$ and $M,w \Vdash_{HT_{LTL}} \neg\psi_k^i$ for every $k \in \{1,\ldots,m\}$ iff $\varphi^i \in T^w$ whenever $\varphi_j^i \in T^w$ for every $j \in \{1,\ldots,n\}$ and $\psi_k^i \notin T^{w'}$ for every $k \in \{1,\ldots,m\}$ and every $w' \geq w$ iff (using Lemma 6) $m^{T^w}, i \Vdash_{LTL} \varphi$ whenever $m^{T^w}, i \Vdash_{LTL} \varphi_j$ for every $j \in \{1,\ldots,n\}$ and $m^{T^{w'}}, i \nVdash_{LTL} \psi_k$ for every $k \in \{1,\ldots,m\}$ and every $w' \geq w$ iff $M^{t2m}, w, i \Vdash_{THT} \varphi$ whenever $M^{t2m}, w, i \Vdash_{THT} \varphi_j$ for every $j \in \{1,\ldots,n\}$ and $M^{t2m}, w, i \Vdash_{THT} \neg\psi_k$ for every $k \in \{1,\ldots,m\}$ iff $M^{t2m}, w, i \Vdash_{THT} (\varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\psi_1 \wedge \ldots \wedge \neg\psi_m) \rightarrow \varphi$ iff $M^{t2m}, w, i \Vdash_{THT} r$. ■

Corollary 2. *Let M be a HT_{LTL} interpretation and P a temporal logic program. Then, $M \Vdash_{HT_{LTL}} P^*$ iff $M^{t2m} \Vdash_{THT} P$.*

Proof. $M^{t2m} \Vdash_{THT} P$ iff $M^{t2m} \Vdash_{THT} \Box r$ for every rule $\Box r \in P$ iff $M^{t2m}, h, 0 \Vdash_{THT} \Box r$ for every rule $\Box r \in P$ iff $M^{t2m}, h, i \Vdash_{THT} r$ for every $i \in \mathbb{N}$ and for every $\Box r \in P$ iff (using Theorem 2) $M, h \Vdash_{HT_{LTL}} r^i$ for every $i \in \mathbb{N}$ and for every rule $\Box r \in P$ iff $M \Vdash_{HT_{LTL}} r^i$ for every $i \in \mathbb{N}$ and for every $\Box r \in P$ iff $M \Vdash_{HT_{LTL}} P^*$. ■

Theorem 3. *Let $M = \langle T^h, T^t \rangle$ be a THT interpretation. Then, for every $w \in \{h,t\}$, every $i \in \mathbb{N}$ and every rule $r = \varphi \leftarrow \varphi_1, \ldots, \varphi_n, not\ \psi_1, \ldots, not\ \psi_m$ we have that $M, w, i \Vdash_{THT} r$ iff $M^{m2t}, w \Vdash_{HT_{LTL}} r^i$.*

Proof. Let $w \in \{h,t\}$ and $i \in \mathbb{N}$. Then $M^{m2t}, w \Vdash_{HT_{LTL}} r^i$ iff $M^{m2t}, w \Vdash_{HT_{LTL}} (\varphi_1^i \wedge \ldots \wedge \varphi_n^i \wedge \neg\psi_1^i \wedge \ldots \wedge \neg\psi_m^i) \rightarrow \varphi^i$ iff $M^{m2t}, w \Vdash_{HT_{LTL}} \varphi^i$ whenever $M^{m2t}, w \Vdash_{HT_{LTL}} \varphi_j^i$ for every $j \in \{1,\ldots,n\}$ and $M^{m2t}, w \Vdash_{HT_{LTL}} \neg\psi_k^i$ for every $k \in \{1,\ldots,m\}$ iff $\varphi^i \in Th(m^w)$ whenever $\varphi_j^i \in Th(m^w)$ for every $j \in \{1,\ldots,n\}$ and $\psi_k^i \notin Th(m^{w'})$ for every $k \in \{1,\ldots,m\}$ and every $w' \geq w$ iff (by definition) $m^w, i \Vdash_{LTL} \varphi$ whenever $m^w, i \Vdash_{LTL} \varphi_j$ for every $j \in \{1,\ldots,n\}$ and $m^{w'}, i \nVdash_{LTL} \psi_k$ for every $k \in \{1,\ldots,m\}$ and every $w' \geq w$ iff $M, w, i \Vdash_{THT} \varphi$ whenever $M, w, i \Vdash_{THT} \varphi_j$ for every $j \in \{1,\ldots,n\}$ and $M, w, i \Vdash_{THT} \neg\psi_k$ for every $k \in \{1,\ldots,m\}$ iff $M, w, i \Vdash_{THT} (\varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\psi_1 \wedge \ldots \wedge \neg\psi_m) \rightarrow \varphi$ iff $M, w, i \Vdash_{THT} r$. ■

Corollary 3. *Let M be a THT interpretation and P be a normal temporal logic program. Then, $M^{m2t} \Vdash_{HT_{LTL}} P^*$ iff $M \Vdash_{THT} P$.*

Proof. $M \Vdash_{THT} P$ iff $M \Vdash_{THT} \Box r$ for every rule $\Box r \in P$ iff $M, h, 0 \Vdash_{THT} \Box r$ for every rule $\Box r \in P$ iff $M, h, i \Vdash_{THT} r$ for every $i \in \mathbb{N}$ and for every rule $\Box r \in P$ iff (using Theorem 3) $M^{m2t}, h \Vdash_{HT_{LTL}} r^i$ for every $i \in \mathbb{N}$ and for every rule $\Box r \in P$ iff $M^{m2t} \Vdash_{HT_{LTL}} P^*$. ■

We are now able to present the main theorem of this section.

Theorem 4. *Let P_1 and P_2 be normal temporal logic programs. Then, P_1 and P_2 are logically equivalent in THT iff P_1^* and P_2^* are logically equivalent in HT_{LTL}.*

Proof. We prove both implication by contraposition. Suppose that P_1 and P_2 are not logically equivalent in THT. Then, without loss of generality, there exists a THT interpretation M such that $M \Vdash_{THT} P_1$ but $M \nVdash_{THT} P_2$. Then, using Corollary 3 we have that $M^{m2t} \Vdash_{HT_{LTL}} P_1^*$ but $M^{m2t} \nVdash_{HT_{LTL}} P_2^*$ and, therefore, P_1^* and P_2^* are not logically equivalent in HT_{LTL}.

Assume now that P_1^* and P_2^* are not logically equivalent in HT_{LTL}. Then, without loss of generality, there exists a HT_{LTL} interpretation M such that $M \Vdash P_1^*$ but $M \nVdash P_2^*$. Then, using Corollary 2 we have that $M^{t2m} \Vdash_{THT} P_1$ but $M^{t2m} \nVdash_{THT} P_2$ and, therefore, P_1 and P_2 are not logically equivalent in THT. ■

We end this section drawing some concluding remarks. As pointed out in [1], logical equivalence in THT is trivially a sufficient condition for strong equivalence. Nevertheless, it is not known whether it is also a necessary condition. Contrarily, in our case, Theorem 1 shows that logical equivalence in HT_{LTL} is a necessary and sufficient condition for strong equivalence of normal temporal logic programs. Moreover, the notion of strong equivalence in [1] is defined using equilibrium models in THT, whereas in our parametrized setting strong equivalence is an usual constructive stable model based notion.

The language of THT is richer than that of HT_{LTL}, in that it allows a full combination of LTL temporal operators and HT connectives. Nevertheless, HT_{LTL} has the advantage of being able to reason about normal temporal logic programs containing classical negation (usually called explicit negation) and classical implication. This is clearly a plus since explicit negation is well known to be a fundamental tool in many reasoning scenarios.

5 Conclusions

We have defined a parametrized version of equilibrium logic and of its monotone base, the logic of Here-and-There, by allowing complex formulas of a parameter logic as atoms. We proved that both Equilibrium logic and HT are particular cases of our approach and, moreover, we generalized the relation between equilibrium logic and the stable model semantics for logic programs. In particular we proved that logical equivalence in parametrized HT captures strong equivalence of parametrized logic programs. We showed that equivalence at the level of the parametrized atoms is compatible with strong equivalence of parametrized logic programs, thus being a first step in the study of simplification of parametrized logic programs. By taking classical logic as parameter logic, we proved that general default logic is a particular case of our approach. We ended with an example where linear temporal logic was taken as parameter logic, thus allowing to characterize strong equivalence of temporal logic programs.

The work raises several interesting paths for future work. One that is already ongoing is to define a parametrized version of partial equilibrium logic and generalise its relation with partial stable model semantics of logic programs. It would be interesting to find an axiomatization of parametrized HT. Moreover,

we would like to extend to this parametrized setting existing techniques for simplifying logic programs. In order to access its merits in full, other interesting examples of parameter logic should be studied. A very interesting example is the case of first-order logic. We intend to compare the resulting parametrized equilibrium logic with the first-order version of equilibrium logic [11].

References

1. F. Aguado, P. Cabalar, G. Pérez, and C. Vidal. Strongly equivalent temporal logic programs. In *Logics in Artificial Intelligence – JELIA*, Lecture Notes in Computer Science, pages 8–20. Springer-Verlag, 2008.
2. Pedro Cabalar. A normal form for linear temporal equilibrium logic. In *Logics in Artificial Intelligence - JELIA*, volume 6341 of *Lecture Notes in Computer Science*, pages 64–76. Springer, 2010.
3. W. A. Carnielli, M. E. Coniglio, D. Gabbay, P. Gouveia, and C. Sernadas. *Analysis and Synthesis of Logics - How To Cut And Paste Reasoning Systems*, volume 35 of *Applied Logic*. Springer, 2008.
4. P. Ferraris. Answer sets for propositional theories. In *Proceedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR*, pages 119–131. Springer, 2005.
5. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *ICLP/SLP*, pages 1070–1080, 1988.
6. R Gonçalves and J. Alferes. Parametrized logic programming. In Tomi Janhunen and Ilkka Niemelä, editors, *Logics in Artificial Intelligence – JELIA*, volume 6341 of *Lecture Notes in Computer Science*, pages 182–194. Springer Berlin / Heidelberg, 2010.
7. V. Lifschitz, D. Pearce, and A. Valverde. Strongly equivalent logic programs. *ACM Trans. Comput. Log.*, 2(4):526–541, 2001.
8. B. Motik and R. Rosati. A faithful integration of description logics with logic programming. In *IJCAI*, pages 477–482, 2007.
9. D. Pearce. Simplifying logic programs under answer set semantics. In *ICLP*, pages 210–224, 2004.
10. D. Pearce. Equilibrium logic. *Ann. Math. Artif. Intell.*, 47(1-2):3–41, 2006.
11. D. Pearce and A. Valverde. Towards a first order equilibrium logic for nonmonotonic reasoning. In José Júlio Alferes and João Alexandre Leite, editors, *Logics in Artificial Intelligence – JELIA*, volume 3229 of *Lecture Notes in Computer Science*, pages 147–160. Springer, 2004.
12. A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (Providence, R.I., 1977)*, pages 46–57. IEEE Comput. Sci., Long Beach, Calif., 1977.
13. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, (13), 1980.
14. R. Wójcicki. *Theory of Logical Calculi*. Synthese Library. Kluwer Academic Publishers, 1988.
15. Y. Zhou, F. Lin, and Y. Zhang. General default logic. *Ann. Math. Artif. Intell.*, 57(2):125–160, 2009.
16. Y. Zhou and Y. Zhang. Rule calculus: Semantics, axioms and applications. In *Logics in Artificial Intelligence – JELIA*, Lecture Notes in Computer Science, pages 416–428. Springer-Verlag, 2008.