Past-present temporal programs over finite traces: a preliminary report

Pedro Cabalar[1], Martín Diéguez[2], François Laferrière[3] and Torsten Schaub[3,4]

[1]University of Corunna, Spain
[2]University of Angers, France
[3]University of Potsdam, Germany
[4]Potassco Solutions, Germany

Abstract

Extensions of Answer Set Programming with language constructs from temporal logics, such as temporal equilibrium logic over finite traces (TEL$_f$), provide an expressive computational framework for modeling dynamic applications. In this paper, we study the so-called past-present syntactic subclass, which consists of a set of logic programming rules whose body references to the past and head to the present. Such restriction ensures that the past remains independent of the future, which is the case in most dynamic domains. We extend the definitions of completion and loop formulas to the case of past-present formulas, which allows capturing the temporal stable models of a set of past-present temporal programs by means of an LTL$_f$ expression.

1. Introduction

The representation and reasoning on dynamic scenarios is considered a central problem in the areas of Knowledge Representation [1] (KR) and Artificial Intelligence (AI). Several formal approaches and systems have emerged in order to introduce non-monotonic reasoning features in scenarios where the formalisation of time is fundamental. Such variety of approaches goes from (non-monotonic) temporal logics [2, 3, 4], to a calculi for reasoning about actions and change [5] or a combination of both approaches [6].

In the area of *Answer Set Programming* [7] (ASP), former approaches to temporal reasoning use first-order encodings [8] where the time is represented by means of a variable whose value comes from a finite domain. The main advantage of those approaches is that the computation of the answer sets of the previous approaches can be achieved via incremental solving [9]. Their downside is that they require an explicit representation of time points. However, their pragmatic advantages, such as a rich (static) modeling language and readily available solvers, so far seem to often outweigh the firm logical foundations of temporal reasoning via modal temporal logics [10].

In order to extend the ASP syntax with connectives from modal temporal logics, more precisely *Linear Time Temporal Logic* [11] (LTL), *Temporal Equilibrium Logic* [12, 13] (TEL) was proposed as a temporal extension of *Equilibrium Logic* [14, 15], the best-known logical characterisation of the stable models semantics for logic programming [16]. Built upon the

ASPOCP'23: 16th Workshop on Answer Set Programming and Other Computing Paradigms, London, UK, July, 2023

product of the logic of here-and-there (HT for short) and LTL, TEL is, to the best our knowledge, the first temporal non-monotonic logic which does not impose any syntactic restriction on the input language.

In the last years, research on TEL shed lights on several of its properties such as complexity [17], expressiveness [18] or computational methods [19, 20].

Due to the computational complexity of its satisfiability problem (ExpSpace), the problem of finding tractable fragments of TEL_f with good computational properties have also been a topic in the literature. Within this context, the so-called *splittable temporal logic programs* [21] have been proved to be a syntactic fragment of TEL that allows a reduction to LTL via the use of Loop Formulas [22].

When focusing on the relation between TEL and incremental solving, temporal logics on finite traces [23] (LTL_f) have been shown to be a suitable semantics for incremental solving of temporal representations. The resulting logic, called *Temporal Equilibrium Logic on Finite traces* [24], became the foundations of the temporal ASP solver *telingo* [25].

In this paper we study a new syntactic fragment of TEL_f, named *past-present* temporal logic programs. Inspired by Gabbay's seminal paper [26], where the declarative character of past temporal operators is emphasized, the language studied in this paper consists of a set of logic programming rules whose formulas in the head are disjunctions of atoms that reference the present, while in its body we allow any arbitrary temporal formula without the use of future operators.

The use of only past operators in the body has the advantage that, when using incremental solving, the body of each rule can be seen as a query whose satisfiability can be checked on the (partial) incremental computation at each step. This advantage allows us to exploit the advantages of *telingo*'s API in order to reuse partial computations during the solving in order increase its performance.

As a contribution, we study the Lin-Zhao theorem [27] within the context of past-present temporal logic programs. More precisely we show that when the program is *tight* [28], extending Clark's completion [29, 30] to the temporal case suffices to capture the answer sets of a finite past-present program Π as the LTL_f-models of a temporal formula φ.

We also show that, when the program is not tight, the use of loop formulas is necessary. To this purpose, we extend the definition of loop formulas to the case of past-present programs and we prove the Lin-Zhao theorem in our setting. Finally, we also prove the generalisation of Lin-Zhao theorem in the sense of [22], where the computation of the completion is be replaced by the consideration of unitary loops.

The paper is organised as follows: in Section 2 we introduce the logic of temporal here and there over finite traces and its equilibrium counterpart. In Section 3 we introduce the concept of past-present temporal programs while, in Section 4 we extend the completion property to the temporal case and we prove that, for the case of tight programs, computing the temporal completion suffices to characterise the temporal answer sets of a past-present programs in terms of LTL_f formulas. The non-tight case is studied in Section 5, where we introduce our temporal extension of loop formulas and extend the results presented in Section 4 to the case of non-tight programs. Our last contribution, presented in Section 6, shows that temporal completion can be captured in the general theory of loop formulas by considering unitary cycles. Finally, in Section 7 we present the conclusions of the paper an we outline some future research lines.

2. Background

For this paper to be self-contained, we first recall the definitions of THT and TEL as in [13]. All logics treated in this paper share the common syntax of LTL with past operators [2]. We start from a given set $\mathcal{A}$ of atoms which we call the *alphabet*. Then, *temporal formulas* φ are defined by the grammar:

$$\varphi ::= a \mid \bot \mid \varphi_1 \otimes \varphi_2 \mid \bullet\varphi \mid \varphi_1 \,\mathbf{S}\, \varphi_2 \mid \varphi_1 \,\mathbf{T}\, \varphi_2 \mid \circ\varphi \mid \varphi_1 \,\mathbb{U}\, \varphi_2 \mid \varphi_1 \,\mathbb{R}\, \varphi_2$$

where $a \in \mathcal{A}$ is an atom and $\otimes$ is any binary Boolean connective $\otimes \in \{\rightarrow, \wedge, \vee\}$. The last six cases correspond to the temporal connectives whose names are listed below:

Past	$\bullet$	for *previous*	*Future*	$\circ$	for *next*
	$\mathbf{S}$	for *since*		$\mathbb{U}$	for *until*
	$\mathbf{T}$	for *trigger*		$\mathbb{R}$	for *release*

the intended meaning of the previous temporal operators is the following: $\bullet\varphi$ (resp. $\circ\varphi$) means that φ is true at the previous (resp. next) time point. $\varphi \,\mathbb{U}\, \psi$ means that φ is true until ψ is true, while $\varphi \,\mathbf{S}\, \psi$ can be read as φ is true since ψ was true. For $\varphi \,\mathbb{R}\, \psi$ and $\varphi \,\mathbf{T}\, \psi$ the meaning is not as direct as for the previous operators. $\varphi \,\mathbb{R}\, \psi$ means that ψ is true until both φ and ψ become true simultaneously or ψ is true forever. $\varphi \,\mathbf{T}\, \psi$ means that ψ is true since both φ and ψ became true simultaneously or ψ has been true from the beginning.

We also define several common derived operators like the Boolean connectives $\top \stackrel{def}{=} \neg\bot$, $\neg\varphi \stackrel{def}{=} \varphi \rightarrow \bot$, $\varphi \leftrightarrow \psi \stackrel{def}{=} (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$, and the following temporal operators:

$\blacksquare\varphi$	$\stackrel{def}{=}$	$\bot \,\mathbf{T}\, \varphi$	*always before*	$\Box\varphi$	$\stackrel{def}{=}$	$\bot \,\mathbb{R}\, \varphi$	*always afterward*
$\blacklozenge\varphi$	$\stackrel{def}{=}$	$\top \,\mathbf{S}\, \varphi$	*eventually before*	$\Diamond\varphi$	$\stackrel{def}{=}$	$\top \,\mathbb{U}\, \varphi$	*eventually afterward*
$\mathbf{I}$	$\stackrel{def}{=}$	$\neg\bullet\top$	*initial*	$\mathbb{F}$	$\stackrel{def}{=}$	$\neg\circ\top$	*final*
$\widehat{\bullet}\varphi$	$\stackrel{def}{=}$	$\bullet\varphi \vee \mathbf{I}$	*weak previous*	$\widehat{\circ}\varphi$	$\stackrel{def}{=}$	$\circ\varphi \vee \mathbb{F}$	*weak next*

A *(temporal) theory* is a (possibly infinite) set of temporal formulas.

Given $a \in \mathbb{N}$ and $b \in \mathbb{N} \cup \{\omega\}$, we let $[a..b]$ stand for the set $\{i \in \mathbb{N} \mid a \leq i \leq b\}$, $[a..b)$ for $\{i \in \mathbb{N} \mid a \leq i < b\}$ and $(a..b]$ for $\{i \in \mathbb{N} \mid a < i \leq b\}$. A *trace* $\mathbf{T}$ of length λ over alphabet $\mathcal{A}$ is a sequence $\mathbf{T} = (T_i)_{i\in[0..\lambda)}$ of sets $T_i \subseteq \mathcal{A}$. We say that $\mathbf{T}$ is *infinite* if $\lambda = \omega$ and *finite* otherwise. To represent a given trace, we write a sequence of sets of atoms concatenated with '$\cdot$'. For instance, the finite trace $\{a\} \cdot \emptyset \cdot \{a\} \cdot \emptyset$ has length 4 and makes a true at even time points and false at odd ones.

A *Here-and-There trace* (for short *HT-trace*) of length λ over alphabet $\mathcal{A}$ is a sequence of pairs $(\langle H_i, T_i\rangle)_{i\in[0..\lambda)}$ with $H_i \subseteq T_i$ for any $i \in [0..\lambda)$. For convenience, we usually represent the HT-trace as the pair $\langle \mathbf{H}, \mathbf{T}\rangle$ of traces $\mathbf{H} = (H_i)_{i\in[0..\lambda)}$ and $\mathbf{T} = (T_i)_{i\in[0..\lambda)}$. Given $\mathbf{M} = \langle \mathbf{H}, \mathbf{T}\rangle$, we also denote its length as $|\mathbf{M}| \stackrel{def}{=} |\mathbf{H}| = |\mathbf{T}| = \lambda$. Note that the two traces $\mathbf{H}$, $\mathbf{T}$ must satisfy a kind of order relation, since $H_i \subseteq T_i$ for each time point i. Formally, we define the ordering $\mathbf{H} \leq \mathbf{T}$ between two traces of the same length λ as $H_i \subseteq T_i$ for each $i \in [0..\lambda)$. Furthermore, we define $\mathbf{H} < \mathbf{T}$ as both $\mathbf{H} \leq \mathbf{T}$ and $\mathbf{H} \neq \mathbf{T}$. Thus, an HT-trace can also be defined as any pair $\langle \mathbf{H}, \mathbf{T}\rangle$ of traces such that $\mathbf{H} \leq \mathbf{T}$. The particular type of HT-traces satisfying $\mathbf{H} = \mathbf{T}$ are called *total*.

Definition 1 (THT-satisfaction; [31, 24]). *An HT-trace $\mathbf{M} = \langle \mathbf{H}, \mathbf{T} \rangle$ of length λ over alphabet $\mathcal{A}$* satisfies *a temporal formula φ at time point $k \in [0..\lambda)$, written $\mathbf{M}, k \models \varphi$, if the following conditions hold:*

1. $\mathbf{M}, k \models \top$ *and* $\mathbf{M}, k \not\models \bot$
2. $\mathbf{M}, k \models p$ *if* $p \in H_k$ *for any atom* $p \in \mathcal{A}$
3. $\mathbf{M}, k \models \varphi \wedge \psi$ *iff* $\mathbf{M}, k \models \varphi$ *and* $\mathbf{M}, k \models \psi$
4. $\mathbf{M}, k \models \varphi \vee \psi$ *iff* $\mathbf{M}, k \models \varphi$ *or* $\mathbf{M}, k \models \psi$
5. $\mathbf{M}, k \models \varphi \to \psi$ *iff* $\langle \mathbf{H}', \mathbf{T} \rangle, k \not\models \varphi$ *or* $\langle \mathbf{H}', \mathbf{T} \rangle, k \models \psi$*, for all* $\mathbf{H}' \in \{\mathbf{H}, \mathbf{T}\}$
6. $\mathbf{M}, k \models \bullet\varphi$ *iff* $k > 0$ *and* $\mathbf{M}, k{-}1 \models \varphi$
7. $\mathbf{M}, k \models \varphi \ \mathbf{S} \ \psi$ *iff for some* $j \in [0..k]$*, we have* $\mathbf{M}, j \models \psi$ *and* $\mathbf{M}, i \models \varphi$ *for all* $i \in (j..k]$
8. $\mathbf{M}, k \models \varphi \ \mathbf{T} \ \psi$ *iff for all* $j \in [0..k]$*, we have* $\mathbf{M}, j \models \psi$ *or* $\mathbf{M}, i \models \varphi$ *for some* $i \in (j..k]$
9. $\mathbf{M}, k \models \circ\varphi$ *iff* $k + 1 < \lambda$ *and* $\mathbf{M}, k{+}1 \models \varphi$
10. $\mathbf{M}, k \models \varphi \ \mathbb{U} \ \psi$ *iff for some* $j \in [k..\lambda)$*, we have* $\mathbf{M}, j \models \psi$ *and* $\mathbf{M}, i \models \varphi$ *for all* $i \in [k..j)$
11. $\mathbf{M}, k \models \varphi \ \mathbb{R} \ \psi$ *iff for all* $j \in [k..\lambda)$*, we have* $\mathbf{M}, j \models \psi$ *or* $\mathbf{M}, i \models \varphi$ *for some* $i \in [k..j)$

□

An HT-trace $\mathbf{M}$ is a *model* of a temporal theory Γ if $\mathbf{M}, 0 \models \varphi$ for all $\varphi \in \Gamma$. We omit specifying LTL satisfaction since it coincides with THT when HT-traces are total.

Proposition 1 ([31, 24]). *Let $\mathbf{T}$ be a trace of length λ, φ a temporal formula, and $k \in [0..\lambda)$ a time point.*

Then, $\mathbf{T}, k \models \varphi$ in LTL iff $\langle \mathbf{T}, \mathbf{T} \rangle, k \models \varphi$. □

Satisfaction of derived operators can be easily deduced, as shown next.

Proposition 2 ([31, 24]). *Let $\mathbf{M} = \langle \mathbf{H}, \mathbf{T} \rangle$ be an HT-trace of length λ over $\mathcal{A}$. Given the respective definitions of derived operators, we get the following satisfaction conditions:*

12. $\mathbf{M}, k \models \mathbf{I}$ *iff* $k = 0$
13. $\mathbf{M}, k \models \widehat{\bullet}\varphi$ *iff* $k = 0$ *or* $\mathbf{M}, k{-}1 \models \varphi$
14. $\mathbf{M}, k \models \blacklozenge\varphi$ *iff* $\mathbf{M}, i \models \varphi$ *for some* $i \in [0..k]$
15. $\mathbf{M}, k \models \blacksquare\varphi$ *iff* $\mathbf{M}, i \models \varphi$ *for all* $i \in [0..k]$
16. $\mathbf{M}, k \models \mathbb{F}$ *iff* $k + 1 = \lambda$
17. $\mathbf{M}, k \models \widehat{\circ}\varphi$ *iff* $k + 1 = \lambda$ *or* $\mathbf{M}, k{+}1 \models \varphi$
18. $\mathbf{M}, k \models \Diamond\varphi$ *iff* $\mathbf{M}, i \models \varphi$ *for some* $i \in [k..\lambda)$
19. $\mathbf{M}, k \models \Box\varphi$ *iff* $\mathbf{M}, i \models \varphi$ *for all* $i \in [k..\lambda)$

□

A formula φ is a *tautology* (or is *valid*), written $\models \varphi$, iff $\mathbf{M}, k \models \varphi$ for any HT-trace $\mathbf{M}$ and any $k \in [0..\lambda)$. We call the logic induced by the set of all tautologies *Temporal logic of Here-and-There* (THT for short).

Proposition 3 (Persistence). *Let $\langle \mathbf{H}, \mathbf{T} \rangle$ be an HT-trace of length λ and φ be a temporal formula. Then, for any $k \in [0..\lambda)$, $k \neq \omega$, if $\langle \mathbf{H}, \mathbf{T} \rangle, k \models \varphi$ then $\langle \mathbf{T}, \mathbf{T} \rangle, k \models \varphi$ (or, if preferred, $\mathbf{T}, k \models \varphi$).* $\square$

As a corollary, we have that $\langle \mathbf{H}, \mathbf{T} \rangle \models \neg\varphi$ iff $\mathbf{T} \not\models \varphi$ in LTL.

Given a set of THT-models, we define the ones in equilibrium as follows.

Definition 2 (Temporal Equilibrium/Stable Model). *Let $\mathfrak{S}$ be some set of HT-traces.*

A total HT-trace $\langle \mathbf{T}, \mathbf{T} \rangle \in \mathfrak{S}$ is a temporal equilibrium model *of $\mathfrak{S}$ iff there is no other $\mathbf{H} < \mathbf{T}$ such that $\langle \mathbf{H}, \mathbf{T} \rangle \in \mathfrak{S}$.*

The trace $\mathbf{T}$ is called a temporal stable model *(*TS-*model) of $\mathfrak{S}$.* $\square$

3. Past-present programs

Given alphabet $\mathcal{A}$, a *pure past formula* φ is defined by the grammar:

$$\varphi ::= a \mid \bot \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \bullet\varphi \mid \varphi_1 \,\mathbf{S}\, \varphi_2 \mid \varphi_1 \,\mathbf{T}\, \varphi_2.$$

where $a \in \mathcal{A}$.

Definition 3 (Past-present Program). *Given alphabet $\mathcal{A}$, the set of* temporal literals *is defined as $\{a, \neg a, \bullet a, \neg\bullet a \mid a \in \mathcal{A}\}$. We refer to its subset $\{a, \neg a \mid a \in \mathcal{A}\}$ as* regular literals*, and $\{\bullet a, \neg\bullet a \mid a \in \mathcal{A}\}$ as* past literals.

A past-present rule *is either:*

- *an* initial rule *of form* $\quad H \leftarrow B$
- *a* dynamic rule *of form* $\quad \widehat{\circ}\,\square(H \leftarrow B)$
- *a* final rule *of form* $\quad \square(\,\mathbb{F} \rightarrow (\bot \leftarrow B)\,)$

where B is an pure past formula for dynamic rules and $B = b_1 \wedge \cdots \wedge b_n$ with $n \geq 0$ for initial and final rules, the b_i are regular literals, $H = a_1 \vee \cdots \vee a_m$ with $m \geq 0$ and $a_j \in \mathcal{A}$. A past-present program *is a set of past-present rules.*

We let $I(P)$, $D(P)$, and $F(P)$ stand for the set of all initial, dynamic, and final rules in a temporal program P, respectively. Additionally we refer to H as the *head* of a rule r and to B as the *body* of r. We let $B(r) = B$ and $H(r) = H$ for all types of rules above.

For example, let consider the following past-present program P_1:

$$load \leftarrow \tag{1}$$

$$\widehat{\circ}\square(shoot \vee load \vee unload \leftarrow\ ) \tag{2}$$

$$\widehat{\circ}\square(dead \leftarrow shoot \wedge \neg unload \,\mathbf{S}\, load) \tag{3}$$

$$\widehat{\circ}\square(shoot \leftarrow dead) \tag{4}$$

$$\square(\mathbb{F} \rightarrow (\bot \leftarrow \neg dead)) \tag{5}$$

We get $I(P_1) = \{(1)\}$, $D(P_1) = \{(2), (3), (4)\}$, and $F(P_1) = \{(3)\}$. Rule (1) states that the gun is initially loaded. Rule (2) give the choice to shoot, load, or unload the gun. Rule (3) states that if the gun is shot while it has been loaded, and not unloaded since, the target is dead. Rule (4) states that if the target is dead, we shoot it again. Rule (5) ensure that the target is dead at the end of the trace. For length $\lambda = 0$, P_1 has a unique TS-model $\{load\} \cdot \{shoot, dead\}$.

4. Temporal completion

Definition 4 (Positive occurence). *An occurrence of an atom in a formula is positive if it is in the antecedent of an even number of implications, negative otherwise.*

Definition 5 (Present occurence). *An occurrence of an atom in a formula is present if it is not in the scope of $\bullet$ (previous).*

Definition 6 (Dependency graph). *Given a past-present program P over $\mathcal{A}$, we define its (positive)* dependency graph *$G(P)$ as $(\mathcal{A}, E)$ such that $(a, b) \in E$ if there is a rule $r \in P$ such that $a \in H(r) \cap \mathcal{A}$ and b has positive and present occurence in $B(r)$ that is not in the scope of negation.*

Definition 7 (Loop). *A nonempty set $L \subseteq \mathcal{A}$ of atoms is called* loop *of P if, for every pair a, b of atoms in L, there exists a path of length > 0 from a to b in $G(P)$ such that all vertices in the path belong to L.*

We let $L(P)$ denote the set of loops of P. Due to the structure of past-present programs, dependencies from the future to the past cannot happen, and therefore there can only be loops within a same time point. To reflect this, the definitions above only consider atoms with present occurences. For example, rule $a \leftarrow b \wedge \bullet c$ generate the edge (a, b) but not (a, c).

Definition 8 (Tight program). *A past-present program P is said to be* tight *if $I(P)$ and $D(P)$ do not contain any loop.*

For P_1, we get for the initial rules

$$\begin{aligned} G(I(P_1)) &= (\{load, unload, shoot, dead\}, \emptyset) \\ L(I(P_1)) &= \emptyset \end{aligned}$$

and for the dynamic rules,

$$\begin{aligned} G(D(P_1)) &= (\{load, unload, shoot, dead\}, \{(dead, shoot), (dead, load), (shoot, dead)\}) \\ L(D(P_1)) &= \{\{shoot, dead\}\} \end{aligned}$$

Definition 9 (Temporal completion). *We define the temporal* completion *formula of an atom a in a past-present program P over $\mathcal{A}$, denoted $CF_P(a)$ as:*

$$\Box\big(a \leftrightarrow \bigvee_{r \in I(P), a \in H(r)} (\mathbf{I} \wedge S(r, a)) \vee \bigvee_{r \in D(P), a \in H(r)} (\neg \mathbf{I} \wedge S(r, a))\big)$$

where $S(r, a) = B(r) \wedge \bigwedge_{p \in H(r) \setminus \{a\}} \neg p$.
The temporal completion *formula of P, denoted $CF(P)$ is*

$$\{CF_P(a) \mid a \in \mathcal{A}\} \cup \{r \mid r \in I(P) \cup D(P), H(r) = \bot\} \cup F(P)$$

Theorem 1. *Lets P be a tight past-present program and $\mathbf{T}$ a trace of length λ. Then, $\mathbf{T}$ is a* TS*-model of P iff $\mathbf{T}$ is a LTL$_f$-model of $CF(P)$.*

The completion of P_1, $CF(P_1)$ is

$$
\begin{aligned}
&\square(load \leftrightarrow \mathbf{I} \vee (\neg\mathbf{I} \wedge \neg shoot \wedge \neg unload))\\
&\square(shoot \leftrightarrow (\neg\mathbf{I} \wedge \neg load \wedge \neg unload)) \vee (\neg\mathbf{I} \wedge dead))\\
&\square(unload \leftrightarrow (\neg\mathbf{I} \wedge \neg shoot \wedge \neg load))\\
&\square(dead \leftrightarrow (\neg\mathbf{I} \wedge shoot \wedge \neg unload\, \mathbf{S}\, load))\\
&\square(\mathbb{F} \rightarrow (\bot \leftarrow \neg dead))
\end{aligned}
$$

For $\lambda = 2$, $CF(P_1)$ has a unique LTL$_f$-model $\{load\} \cdot \{shoot, dead\}$, which is identical to the TS-model of P_1. Notice that for this example, the TS-models of the program match the LTL$_f$-models of its completion despite the program not being tight. It is generally not the case. Let P_2 be the program made of the rules (1), (3), (4) and (5). The completion of P_2, $CF(P_2)$ is

$$
\begin{aligned}
&\square(load \leftrightarrow \mathbf{I})\\
&\square(shoot \leftrightarrow (\neg\mathbf{I} \wedge dead))\\
&\square(unload \leftrightarrow \bot)\\
&\square(dead \leftrightarrow (\neg\mathbf{I} \wedge shoot \wedge \neg unload\, \mathbf{S}\, load))\\
&\square(\mathbb{F} \rightarrow (\bot \leftarrow \neg dead))
\end{aligned}
$$

P_2 does not have any TS-model, but $\{load\} \cdot \{shoot, dead\}$ is a LTL$_f$-model of $CF(P_2)$. Under ASP semantic, it is impossible to derive any element of the loop $\{shoot, dead\}$, as deriving $dead$ requires $shoot$ to be true, and deriving $shoot$ requires $dead$ to be true. The completion does not restrict this kind of circular derivation and therefore is insufficient to fully capture ASP semantic.

5. Temporal loop formulas

Lin and Zhao [27] introduced the concept of loop formulas to restrict circular derivations. In this section, we extend the notion of loop formula to past-present programs.

Definition 10. *Let φ be a implication-free past-present formula and L a loop. We define the supporting transformation of φ with respect to L as*

$$
\begin{aligned}
S_\bot(L) &\stackrel{def}{=} \bot\\
S_p(L) &\stackrel{def}{=} \begin{cases}\bot & \text{if } p \in L\\ p & \text{otherwise}\end{cases} \qquad \textit{for any atom } p \in \mathcal{A}\\
S_{\neg\varphi}(L) &\stackrel{def}{=} \neg\varphi\\
S_{\varphi\wedge\psi}(L) &\stackrel{def}{=} S_\varphi(L) \wedge S_\psi(L)
\end{aligned}
$$

$$
\begin{aligned}
S_{\varphi\vee\psi}(L) &\stackrel{def}{=} S_\varphi(L) \vee S_\psi(L)\\
S_{\bullet\varphi}(L) &\stackrel{def}{=} \bullet\varphi\\
S_{\varphi\,\mathbf{T}\,\psi}(L) &\stackrel{def}{=} S_\psi(L) \wedge (S_\varphi(L) \vee \bullet(\varphi\ \mathbf{T}\ \psi))\\
S_{\varphi\,\mathbf{S}\,\psi}(L) &\stackrel{def}{=} S_\psi(L) \vee (S_\varphi(L) \wedge \bullet(\varphi\ \mathbf{S}\ \psi))
\end{aligned}
$$

Definition 11 (External support). *Given a past-present program P, the external support formula of a set of atoms $L \subseteq \mathcal{A}$ wrt P, is defined as*

$$ES_P(L) = \bigvee_{r\in P, H(r)\cap L\neq\emptyset} \big(S_{B(r)}(L) \wedge \bigwedge_{a\in H(r)\setminus L} \neg a\big)$$

For our examples P_1 and P_2, and $L = \{\mathit{shoot}, \mathit{dead}\}$, we have

$$
\begin{aligned}
ES_{P_2}(L) &= S_{\mathit{dead}}(L) \vee S_{\mathit{shoot}\wedge\neg \mathit{unload}\,\mathbf{S}\,\mathit{load}}(L)\\
&= S_{\mathit{dead}}(L) \vee (S_{\mathit{shoot}}(L) \wedge S_{\neg \mathit{unload}\,\mathbf{S}\,\mathit{load}}(L))\\
&= S_{\mathit{dead}}(L) \vee (S_{\mathit{shoot}}(L) \wedge S_{\neg \mathit{unload}}(L) \vee \bullet(\neg \mathit{unload}\ \mathbf{S}\ \mathit{load}))\\
&= \bot \vee (\bot \wedge \neg \mathit{unload} \vee \bullet(\neg \mathit{unload}\ \mathbf{S}\ \mathit{load}))\\
&= \bot
\end{aligned}
$$

and

$$
\begin{aligned}
ES_{P_1}(L) &= S_{\mathit{dead}}(L) \vee S_{\mathit{shoot}\wedge\neg \mathit{unload}\,\mathbf{S}\,\mathit{load}}(L) \vee (\neg \mathit{load} \wedge \neg \mathit{unload})\\
&= \neg \mathit{load} \wedge \neg \mathit{unload}
\end{aligned}
$$

Rule (2) provide an external support for L. The body dead of rule (4) is also a support for L, but not external as dead belongs to L. The supporting transformation only keeps external supports by removing from the body any positive and present occurence of element of L.

Definition 12 (Loop formulas). *We define the set of loop formulas of a past-present program P over $\mathcal{A}$, denoted $LF(P)$, as:*

$$\bigvee_{a\in L} a \rightarrow ES_{I(P)}(L) \text{ for any loop } L \text{ in } I(P) \tag{6}$$

$$\widehat{\circ}\Box\Big(\bigvee_{a\in L} a \rightarrow ES_{D(P)}(L)\Big) \text{ for any loop } L \text{ in } D(P) \tag{7}$$

Theorem 2. *Let P be a past-present program and $\mathbf{T}$ a trace of length λ. Then, $\mathbf{T}$ is a TS-model of P iff $\mathbf{T}$ is a LTL$_f$-model of $CF(P) \cup LF(P)$.*

For our examples, we have

$$LF(P_1) = \widehat{\circ}\Box(\mathit{shoot} \vee \mathit{dead} \rightarrow \neg \mathit{load} \wedge \neg \mathit{unload})$$

and

$$LF(P_2) = \widehat{\circ}\Box(\mathit{shoot} \vee \mathit{dead} \rightarrow \bot)$$

$\{\mathit{load}\}\cdot\{\mathit{shoot}, \mathit{dead}\}$ satisfies $LF(P_1)$, but not $LF(P_2)$. So, we have that $CF(P_1)\cup LF(P_1)$ has a unique LTL$_f$-model $\{\mathit{load}\}\cdot\{\mathit{shoot}, \mathit{dead}\}$, while $CF(P_2)\cup LF(P_2)$ has no LTL$_f$-model, matching the TS-models of the respective programs.

6. Temporal loop formulas with unitary cycles

Ferraris et al. [22] proposed an approach where the computation of the completion can be avoided by considering unitary cycles. In this section, we extend such results for past-present programs. We first redefine loops so that unitary cycles are included.

Definition 13 (Unitary cycle). *A nonempty set $L \subseteq \mathcal{A}$ of atoms is called* loop *of P if, for every pair a, b of atoms in L, there exists a path (possibly of length 0) from a to b in $G(P)$ such that all vertices in the path belong to L.*

With this definition, it is clear that any set consisting of a single atom is a loop. For example, $L(D(P_1)) = \{\{load\}, \{unload\}, \{shoot\}, \{dead\}, \{shoot, dead\}\}$.

Theorem 3. *Let P be a past-present program and $\mathbf{T}$ a trace of length λ. Then, $\mathbf{T}$ is a TS-model of P iff $\mathbf{T}$ is a LTL$_f$-model of $P \cup LF(P)$.*

With unitary cycle, $LF(P_1)$ becomes

$$
\begin{aligned}
load &\rightarrow \top \\
unload &\rightarrow \bot \\
shoot &\rightarrow \bot \\
dead &\rightarrow \bot \\
\widehat{\circ}\Box(load &\rightarrow \neg shoot \wedge \neg unload) \\
\widehat{\circ}\Box(unload &\rightarrow \neg shoot \wedge \neg load) \\
\widehat{\circ}\Box(shoot &\rightarrow (\neg shoot \wedge \neg load) \vee dead) \\
\widehat{\circ}\Box(dead &\rightarrow shoot \wedge \neg unload \,\mathbf{S}\, load) \\
\widehat{\circ}\Box(shoot \vee dead &\rightarrow \neg load \wedge \neg unload)
\end{aligned}
$$

$P_1 \cup LF(P_1)$ has the same LTL$_f$-model $CF(P_1) \cup LF(P_1)$, $\{load\} \cdot \{shoot, dead\}$, which is the TS-model of P_1.

7. Conclusions and Future Work

In this paper we have focused on the computation methods for temporal logic programming within the context of Temporal Equilibrium Logic over finite traces. More precisely, we have studied a fragment close to logic programming rules in the spirit of [26]: a past-present temporal logic program consists of a set of rules whose body refers to the past and present while their head refers to the present. This fragment is very interesting for implementation purposes since it can be solved by means of incremental solving techniques like those implemented in *telingo*. Moreover, restricting the body of the rules to only present and past formulas makes it so that the body of the rules can be seen as queries on the set of conclusions generated during the solving phase.

Contrary to the propositional case [22], where the answer sets of an arbitrary propositional formula can be captures by means of the classical models of another formula ψ, in the temporal case this is not possible to do the same mapping among the temporal equilibrium models of a formula φ and the LTL models of another formula ψ [18].

In this paper we show that past-present temporal logic programs can be effectively reduced to LTL formulas by means of completion and loop formulas. More precisely we extend the definition of completion and temporal loop formulas in the spirit of Lin and Zhao [27] to the temporal case, and we show that for tight past-present programs, the use of completion is sufficient to achieve a reduction to an LTL_f formula. Moreover, when the program is not tight, we also show that the computation of the temporal completion and a finite number of loop formulas suffices to reduce TEL_f to LTL_f. Finally, we consider Ferraris et al. approach [22] where the computation of the completion can be automatically replaced by the consideration of unitary loops. In this contribution, we extend such result to the case of past-present logic programs.

As future work we plan to study in detail the relation between temporal completion and loop formulas and *unfounded sets* [22, 32], since the latter plays a central role in the solving algorithm of *clingo* [33, 34]. Lastly, we will study the case of past-present temporal programs with variables [35] in order to get a second-order characterisation of loop formulas, in the spirit of [36].

References

[1] R. J. Brachman, H. J. Levesque, Knowledge Representation and Reasoning, Elsevier, 2004. URL: `http://www.elsevier.com/wps/find/bookdescription.cws_home/702602/description`.

[2] E. Emerson, Temporal and modal logic, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science, MIT Press, 1990, pp. 995–1072.

[3] C. Baral, J. Zhao, Non-monotonic temporal logics that facilitate elaboration tolerant revision of goals, in: D. Fox, C. P. Gomes (Eds.), Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008, AAAI Press, 2008, pp. 406–411.

[4] C. Baral, J. Zhao, Non-monotonic temporal logics for goal specification, in: M. M. Veloso (Ed.), IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007, 2007, pp. 236–242.

[5] E. Sandewall, Features and fluents: the representation of knowledge about dynamical systems, volume 1, Oxford University Press, New York, NY, USA, 1994.

[6] G. González, C. Baral, P. A. Cooper, Modeling Multimedia Displays Using Action Based Temporal Logic, Springer US, Boston, MA, 2002, pp. 141–155.

[7] G. Brewka, T. Eiter, M. Truszczyński, Answer set programming at a glance, Communications of the ACM 54 (2011) 92–103.

[8] V. Lifschitz, Answer set planning, in: D. de Schreye (Ed.), Proceedings of the International Conference on Logic Programming (ICLP'99), MIT Press, 1999, pp. 23–37.

[9] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, S. Thiele, Engineering an incremental ASP solver, in: M. Garcia de la Banda, E. Pontelli (Eds.), Proceedings of the

Twenty-fourth International Conference on Logic Programming (ICLP'08), volume 5366 of *Lecture Notes in Computer Science*, Springer-Verlag, 2008, pp. 190–205.

[10] S. Demri, V. Goranko, M. Lange, Temporal Logics in Computer Science: Finite-State Systems, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2016.

[11] A. Pnueli, The temporal logic of programs, in: Proceedings of the Eight-teenth Symposium on Foundations of Computer Science (FOCS'77), IEEE Computer Society Press, 1977, pp. 46–57.

[12] P. Cabalar, G. P. Vega, Temporal equilibrium logic: A first approach, in: R. Moreno-Díaz, F. Pichler, A. Quesada-Arencibia (Eds.), Proceedings of the Eleventh International Conference on Computer Aided Systems Theory (EUROCAST'17), volume 4739 of *Lecture Notes in Computer Science*, Springer-Verlag, 2007, pp. 241–248.

[13] F. Aguado, P. Cabalar, M. Diéguez, G. Pérez, T. Schaub, A. Schuhmann, C. Vidal, Linear-time temporal answer set programming, Theory Pract. Log. Program. 23 (2023) 2–56.

[14] D. Pearce, A new logical characterisation of stable models and answer sets, in: J. Dix, L. Pereira, T. Przymusinski (Eds.), Proceedings of the Sixth International Workshop on Non-Monotonic Extensions of Logic Programming (NMELP'96), volume 1216 of *Lecture Notes in Computer Science*, Springer-Verlag, 1997, pp. 57–70.

[15] D. Pearce, Equilibrium logic, Annals of Mathematics and Artificial Intelligence 47 (2006) 3–41.

[16] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming, in: R. Kowalski, K. Bowen (Eds.), Proceedings of the Fifth International Conference and Symposium of Logic Programming (ICLP'88), MIT Press, 1988, pp. 1070–1080.

[17] L. Bozzelli, D. Pearce, On the complexity of temporal equilibrium logic, in: Proceedings of the Thirtieth Annual Symposium on Logic in Computer Science (LICS'15), IEEE Computer Society Press, 2015, pp. 645–656.

[18] L. Bozzelli, D. Pearce, On the expressiveness of temporal equilibrium logic, in: L. Michael, A. Kakas (Eds.), Proceedings of the Fifteenth European Conference on Logics in Artificial Intelligence (JELIA'16), volume 10021 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 2016, pp. 159–173.

[19] P. Cabalar, M. Diéguez, STELP — a tool for temporal answer set programming, in: [37], 2011, pp. 370–375.

[20] P. Cabalar, S. Demri, Automata-based computation of temporal equilibrium models, in: G. Vidal (Ed.), Proceedings of the Twenty-first International Symposium on Logic-Based Program Synthesis and Transformation (LOPSTR'11), volume 7225 of *Lecture Notes in Computer Science*, Springer-Verlag, 2011, pp. 57–72.

[21] F. Aguado, P. Cabalar, G. Pérez, C. Vidal, Loop formulas for splitable temporal logic programs, in: [37], 2011, pp. 80–92.

[22] P. Ferraris, J. Lee, V. Lifschitz, A generalization of the Lin-Zhao theorem, Annals of Mathematics and Artificial Intelligence 47 (2006) 79–101.

[23] G. De Giacomo, M. Vardi, Linear temporal logic and linear dynamic logic on finite traces, in: F. Rossi (Ed.), Proceedings of the Twenty-third International Joint Conference on Artificial Intelligence (IJCAI'13), IJCAI/AAAI Press, 2013, pp. 854–860.

[24] P. Cabalar, R. Kaminski, T. Schaub, A. Schuhmann, Temporal answer set programming on

finite traces, Theory and Practice of Logic Programming 18 (2018) 406–420.

[25] P. Cabalar, R. Kaminski, P. Morkisch, T. Schaub, telingo = ASP + time, in: M. Balduccini, Y. Lierler, S. Woltran (Eds.), Proceedings of the Fifteenth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'19), volume 11481 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 2019, pp. 256–269.

[26] D. Gabbay, The declarative past and imperative future: Executable temporal logic for interactive systems, in: B. Banieqbal, H. Barringer, A. Pnueli (Eds.), Proceedings of the Conference on Temporal Logic in Specification, volume 398 of *Lecture Notes in Computer Science*, Springer-Verlag, 1987, pp. 409–448.

[27] F. Lin, J. Zhao, On tight logic programs and yet another translation from normal logic programs to propositional logic, in: G. Gottlob, T. Walsh (Eds.), Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03), Morgan Kaufmann Publishers, 2003, pp. 853–858.

[28] E. Erdem, V. Lifschitz, Tight logic programs, Theory and Practice of Logic Programming 3 (2003) 499–518.

[29] K. Clark, Negation as failure, in: H. Gallaire, J. Minker (Eds.), Logic and Data Bases, Plenum Press, 1978, pp. 293–322.

[30] F. Fages, Consistency of Clark's completion and the existence of stable models, Journal of Methods of Logic in Computer Science 1 (1994) 51–60.

[31] F. Aguado, P. Cabalar, M. Diéguez, G. Pérez, C. Vidal, Temporal equilibrium logic: a survey, Journal of Applied Non-Classical Logics 23 (2013) 2–24.

[32] V. Lifschitz, Answer Set Programming, Springer, 2019.

[33] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, *Clingo* = ASP + Control: Extended Report, Technical Report, Universität Potsdam, 2014. URL: `http://www.cs.uni-potsdam.de/wv/pdfformat/gekakasc14a.pdf`.

[34] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, *Clingo* = ASP + control: Preliminary report, in: M. Leuschel, T. Schrijvers (Eds.), Technical Communications of the Thirtieth International Conference on Logic Programming (ICLP'14), volume 14(4-5) of *Theory and Practice of Logic Programming, Online Supplement*, 2014. Available at `http://arxiv.org/abs/1405.3694v1`.

[35] F. Aguado, P. Cabalar, G. Pérez, C. Vidal, M. Diéguez, Temporal logic programs with variables, Theory Pract. Log. Program. 17 (2017) 226–243. URL: `https://doi.org/10.1017/S1471068416000570`. doi:`10.1017/S1471068416000570`.

[36] J. Lee, Y. Meng, On loop formulas with variables, in: G. Brewka, J. Lang (Eds.), Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR'08), AAAI Press, 2008, pp. 444–453.

[37] J. Delgrande, W. Faber (Eds.), Proceedings of the Eleventh International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'11), volume 6645 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 2011.

A. Proofs (for reviewing purposes)

Definition 14. *Let $\langle \mathbf{H}, \mathbf{T} \rangle$ and $\langle \mathbf{H}', \mathbf{T} \rangle$ be two HT-traces of length λ and let $i \in [0..\lambda)$. We say denote by $\langle \mathbf{H}, \mathbf{T} \rangle \sim_i \langle \mathbf{H}', \mathbf{T} \rangle$ the fact for all $j \in [0..i)$, $H_i = H_i'$.*

Proposition 4. *For all HT-traces $\langle \mathbf{H}, \mathbf{T} \rangle$ and $\langle \mathbf{H}', \mathbf{T} \rangle$ and for all $i \in [0..\lambda)$, if $\langle \mathbf{H}, \mathbf{T} \rangle \sim_i \langle \mathbf{H}', \mathbf{T} \rangle$ then for all $j \in [0..i)$ and for all past formulas φ, $\langle \mathbf{H}, \mathbf{T} \rangle, j \models \varphi$ iff $\langle \mathbf{H}', \mathbf{T} \rangle, j \models \varphi$*

Definition 15 ($\mathbf{X}^i$). *Let $\langle \mathbf{H}, \mathbf{T} \rangle$ be a HT-trace of length λ and $i \in [0..\lambda)$. We denote $\mathbf{X}^i$ the trace of length λ satisfying $X^i_k = \emptyset$ for all $k \in [0..i)$.*

Lemma 1. *For all HT-traces $\langle \mathbf{H}, \mathbf{T} \rangle$ of length λ, for all $i \in [0..\lambda)$ and for any pure past formula φ, if each present and positive occurrence of an atom from X^i_i in φ is in the scope of negation then $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \varphi$ iff $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle, i \models \varphi$.*

Proof of Lemma 1. By induction on φ. First, note that for any formula ϕ of the form $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \,\mathsf{T}\, \psi$ or $\varphi \,\mathsf{S}\, \psi$, if all present and positive occurrences of an atom p are in the scope of negation in ϕ, then all present and positive occurrences of p are also in the scope of negation in φ and ψ.

- case $\bot$: clearly, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models \bot$ and $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle, i \not\models \bot$.
- case p: we consider two cases. If $p \notin X^i_i$, $\langle \mathbf{H}, \mathbf{T} \rangle, i \models p$ iff $p \in H_i$, iff $p \in H_i \setminus X_i$, iff $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle, i \models p$.
  If $p \in X^i_i$, then p has a present and positive occurence in φ, which is not in the scope of negation. Therefore, the lemma automatically holds.
- case $\neg\varphi$: $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \neg\varphi$ iff $\langle \mathbf{T}, \mathbf{T} \rangle, i \not\models \varphi$ iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T} \rangle, i \models \neg\varphi$ (because of persistency).
- case $\bullet\varphi$:
  - if $i = 0$, then $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models \bullet\varphi$ and $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle, i \not\models \bullet\varphi$.
  - if $i > 0$, then $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \bullet\varphi$ iff $\langle \mathbf{H}, \mathbf{T} \rangle, i-1 \models \varphi$. Let X^{i-1} be such that $X^{i-1}_{i-1} = \emptyset$. Then, we can apply the induction hypothesis, so $\langle \mathbf{H}, \mathbf{T} \rangle, i-1 \models \varphi$ iff (IH) $\langle \mathbf{H} \setminus \mathbf{X}^{i-1}, \mathbf{T} \rangle, i-1 \models \varphi$. Since $\langle \mathbf{H} \setminus \mathbf{X}^{i-1}, \mathbf{T} \rangle \sim_i \langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle$ (Proposition 4) then $\langle \mathbf{H} \setminus \mathbf{X}^{i-1}, \mathbf{T} \rangle, i-1 \models \varphi$ iff $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle, i-1 \models \varphi$, iff $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle, i \models \bullet\varphi$.
- case $\varphi \vee \psi$: $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \varphi \vee \psi$ iff $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \varphi$ or $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \psi$. Since all positive and present occurences of atoms from X^i_i in φ and ψ are in the scope of negation, we can apply the induction hypothesis to get $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle, i \models \varphi$ or $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle, i \models \psi$. Therefore, $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle, i \models \varphi \vee \psi$.
- case $\varphi \wedge \psi$: Similar as for $\varphi \vee \psi$.
- case $\varphi \,\mathsf{S}\, \psi$: $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \varphi \,\mathsf{S}\, \psi$ iff for some $j \in [0..i]$, $\langle \mathbf{H}, \mathbf{T} \rangle, j \models \psi$ and $\langle \mathbf{H}, \mathbf{T} \rangle, k \models \varphi$ for all $k \in (j..i]$. By induction we get that iff for some $j \in [0..i]$, $\langle \mathbf{H} \setminus \mathbf{X}^j, \mathbf{T} \rangle, j \models \psi$ and $\langle \mathbf{H} \setminus \mathbf{X}^k, \mathbf{T} \rangle, k \models \varphi$ for all $k \in (j..i]$. Since $\langle \mathbf{H} \setminus \mathbf{X}^t, \mathbf{T} \rangle \sim_t \langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle$ for all $t \in [0..i)$, by Proposition 4 we get that iff for some $j \in [0..i]$, $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle, j \models \psi$ and $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle, k \models \varphi$ for all $k \in (j..i]$. iff $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T} \rangle, i \models \varphi \,\mathsf{S}\, \psi$.

- case $\varphi \mathbf{T} \psi$: assume by contradiction that $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T}\rangle, i \not\models \varphi \mathbf{T} \psi$. This means that there exist $j \in [0..i]$ such that $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T}\rangle, j \not\models \psi$ and $\langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T}\rangle, k \not\models \varphi$ for all $k \in (j..i]$. Since $\langle \mathbf{H} \setminus \mathbf{X}^t, \mathbf{T}\rangle \sim_t \langle \mathbf{H} \setminus \mathbf{X}^i, \mathbf{T}\rangle$ for all $t \in [0..i)$, by Proposition 4 we get that there exist $j \in [0..i]$ such that $\langle \mathbf{H} \setminus \mathbf{X}^j, \mathbf{T}\rangle, j \not\models \psi$ and $\langle \mathbf{H} \setminus \mathbf{X}^k, \mathbf{T}\rangle, k \not\models \varphi$ for all $k \in (j..i]$. By induction, there exist $j \in [0..i]$ such that $\langle \mathbf{H}, \mathbf{T}\rangle, j \not\models \psi$ and $\langle \mathbf{H}, \mathbf{T}\rangle, k \not\models \varphi$ for all $k \in (j..i]$ iff $\langle \mathbf{H}, \mathbf{T}\rangle, i \not\models \varphi \mathbf{T} \psi$: a contradiction.

$\square$

Definition 16. *Let $L \subseteq \mathcal{A}$ and let $\lambda > 0$ and $i \in [0..\lambda)$. By $\mathbf{X}(L)^i$ we mean a trace of length λ satisfying the following conditions:*

1. $L \subseteq X(L)^i_i \subseteq \mathcal{A}$*;*
2. $X(L)^i_t = \emptyset$ *for all* $t \in [0..i)$*.*

Lemma 2. *Let $\langle \mathbf{H}, \mathbf{T}\rangle$ be a HT-trace of length λ, φ an pure past formula. Let us consider the set of atoms $L \subseteq \mathcal{A}$. For all $i \in [0..\lambda)$, if each positive occurrence of an atom from $X(L)^i_i \setminus L$ in φ is in the scope of negation, $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_\varphi(L)$ iff $\langle \mathbf{H} \setminus \mathbf{X}(L)^i, \mathbf{T}\rangle, i \models \varphi$.*

Proof of Lemma 2. By induction on φ. First, note that for any formula ϕ of the form $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \mathbf{T} \psi$ or $\varphi \mathbf{S} \psi$, if all present and positive occurrences of an atom p are in the scope of negation in ϕ, then all present and positive occurrences of p are also in the scope of negation in φ and ψ.

- case $\bot$: $\langle \mathbf{H}, \mathbf{T}\rangle, i \not\models \bot$ and $\langle \mathbf{H} \setminus \mathbf{X}(L)^i, \mathbf{T}\rangle, i \not\models \bot$.
- case $p \notin L$: we consider the following two cases
  - If $p \notin L$, $S_p(L) = p$ and, by definition, $p \notin X(L)^i_i$. Therefore, $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_p(L)$ iff $\langle \mathbf{H}, \mathbf{T}\rangle, i \models p$, iff $p \in H_i$ iff $p \in H_i \setminus X(L)^i_i$, iff $\langle \mathbf{H} \setminus \mathbf{X}(L)^i, \mathbf{T}\rangle, i \models p$.
  - If $p \in L$ then $p \notin X(L)^i_i \setminus L$ and $S_p(L) = \bot$. Therefore, we get that $\langle \mathbf{H}, \mathbf{T}\rangle, i \not\models S_p(L)$ and $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \not\models p$.
- case $\neg\varphi$: $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_{\neg\varphi}(L)$ iff $\langle \mathbf{H}, \mathbf{T}\rangle, i \models \neg\varphi$, iff $\langle \mathbf{T}, \mathbf{T}\rangle, i \not\models \varphi$, iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \neg\varphi$.
- case $\bullet\varphi$: $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_{\bullet\varphi}(L)$ iff $\langle \mathbf{H}, \mathbf{T}\rangle, i \models \bullet\varphi$.
  - If $i = 0$, then both $\langle \mathbf{H}, \mathbf{T}\rangle, 0 \not\models \bullet\varphi$ and $\langle \mathbf{H} \setminus \mathbf{X}(L)^0, \mathbf{T}\rangle, 0 \not\models \bullet\varphi$.
  - If $i > 0$, then $\langle \mathbf{H}, \mathbf{T}\rangle, i \models \bullet\varphi$ iff $\langle \mathbf{H}, \mathbf{T}\rangle, i-1 \models \varphi$. By definition, $X(L)^i_j = \emptyset$ for $j \in [0..i)$ so by Lemma 1, $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i-1 \models \varphi$, iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \bullet\varphi$.
- case $\varphi \wedge \psi$: $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_{\varphi\wedge\psi}(L)$
  iff $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_\varphi(L)$ and $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_\psi(L)$,
  iff (IH) $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \varphi$ and $\langle \mathbf{H} \setminus \mathbf{X}^i(X), \mathbf{T}\rangle, i \models \psi$,
  iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \varphi \wedge \psi$.
- case $\varphi \vee \psi$: $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_{\varphi\vee\psi}(L)$
  iff $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_\varphi(L)$ or $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_\psi(L)$,
  iff (IH) $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \varphi$ or $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \psi$,
  iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \varphi \vee \psi$.

- case $\varphi \,\mathsf{S}\, \psi$: $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_{\varphi \mathsf{S} \psi}(L)$ iff
  1. $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_{\psi}(L)$ iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \psi$ (IH) or
  2. $\langle \mathbf{H}, \mathbf{T}\rangle, i \models S_{\varphi}(L)$ and $\langle \mathbf{H}, \mathbf{T}\rangle, i \models \bullet(\varphi \,\mathsf{S}\, \psi)$ iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \varphi$ (IH) and $\langle \mathbf{H}, \mathbf{T}\rangle, i \models \bullet(\varphi \,\mathsf{S}\, \psi)$ iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \varphi$ and $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \bullet(\varphi \,\mathsf{S}\, \psi)$

  From the previous items we conclude iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \psi \vee (\varphi \wedge \bullet (\varphi \,\mathsf{S}\, \psi))$ iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \models \varphi \,\mathsf{S}\, \psi$.
- case $\varphi \,\mathsf{T}\, \psi$: $\langle \mathbf{H}, \mathbf{T}\rangle, i \not\models S_{\varphi \mathsf{T} \psi}(L)$ iff
  1. $\langle \mathbf{H}, \mathbf{T}\rangle, i \not\models S_{\psi}(L)$ iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \not\models \psi$ (IH) or
  2. $\langle \mathbf{H}, \mathbf{T}\rangle, i \not\models S_{\varphi}(L)$ and $\langle \mathbf{H}, \mathbf{T}\rangle, i \not\models \bullet(\varphi \,\mathsf{T}\, \psi)$ iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \not\models \varphi$ (IH) and $\langle \mathbf{H}, \mathbf{T}\rangle, i \not\models \bullet(\varphi \,\mathsf{T}\, \psi)$ iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \not\models \varphi$ and $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \not\models \bullet(\varphi \,\mathsf{T}\, \psi)$

  From the previous items we conclude iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \not\models \psi \wedge (\varphi \vee \bullet (\varphi \,\mathsf{T}\, \psi))$ iff $\langle \mathbf{H} \setminus \mathbf{X}^i(L), \mathbf{T}\rangle, i \not\models \varphi \,\mathsf{T}\, \psi$.

$\square$

Proof of Theorem 1. From left to right, let us assume towards a contradiction that $\mathbf{T}$ is a temporal answer set of P, but $\mathbf{T}$ is not an LTL$_f$-model of $CF(P)$. By construction, if $\mathbf{T}$ is a temporal answer set of P then $\mathbf{T}$ is an LTL$_f$ model of P so $\mathbf{T}, 0 \models P$. Therefore, $\mathbf{T}, 0 \models r$, for all $r \in P$ such that $H(r) = \bot$ and $\mathbf{T}, 0 \models r$ for all $r \in F(P)$. Since $\mathbf{T}, 0 \not\models CF(P)$, there exists $a \in \mathcal{A}$ such that

$$\mathbf{T}, 0 \not\models \Box\big(a \leftrightarrow \bigvee_{r \in I(P), a \in H(r)} (\mathsf{I} \wedge S(r,a)) \vee \bigvee_{r \in D(P), a \in H(r)} (\neg \mathsf{I} \wedge S(r,a))\big)$$

So, there exists $i \in [0..\lambda)$ such that

$$\mathbf{T}, i \not\models a \leftrightarrow \bigvee_{r \in I(P), a \in H(r)} (\mathsf{I} \wedge S(r,a)) \vee \bigvee_{r \in D(P), a \in H(r)} (\neg \mathsf{I} \wedge S(r,a))$$

We consider two cases:

1. $\mathbf{T}, i \models a$ and $\mathbf{T}, i \not\models \bigvee_{r \in I(P), a \in H(r)} (\mathsf{I} \wedge S(r,a)) \vee \bigvee_{r \in D(P), a \in H(r)} (\neg \mathsf{I} \wedge S(r,a))$:
   - If $i = 0$ then we get that for all $r \in I(P)$, if $a \in H(r)$ then $\langle \mathbf{T}, \mathbf{T}\rangle, 0 \not\models S(r,a)$. Therefore, for all $r \in I(P)$, if $a \in H(r)$ then $\langle \mathbf{T}, \mathbf{T}\rangle, 0 \not\models \S(r,a)$. Let $\mathbf{H}$ be a trace of length λ such that $H_0 = T_0 \setminus \{a\}$ and $H_i = T_i$ for $i \in [1..\lambda)$. Clearly, $\mathbf{H} < \mathbf{T}$. We show a contradiction by proving that $\langle \mathbf{H}, \mathbf{T}\rangle \models P$:
     a) $\langle \mathbf{H}, \mathbf{T}\rangle, 0 \models I(P)$: note that $\langle \mathbf{T}, \mathbf{T}\rangle, 0 \models B(r) \rightarrow H(r)$ for all $r \in I(P)$, iff for any $r \in I(P)$, $\langle \mathbf{T}, \mathbf{T}\rangle, 0 \not\models B(r)$ or $\langle \mathbf{T}, \mathbf{T}\rangle, 0 \models H(r)$. If $\langle \mathbf{T}, \mathbf{T}\rangle, 0 \not\models B(r)$ then, by persistence, $\langle \mathbf{H}, \mathbf{T}\rangle, 0 \not\models B(r)$, and $\langle \mathbf{H}, \mathbf{T}\rangle, 0 \models B(r) \rightarrow H(r)$. If $\langle \mathbf{T}, \mathbf{T}\rangle, 0 \models B(r)$, then $\langle \mathbf{T}, \mathbf{T}\rangle, 0 \models H(r)$. There are two cases.
        - Case $a \notin H(r)$: $\langle \mathbf{T}, \mathbf{T}\rangle, 0 \models H(r)$ so there is some $p \in H(r)$ such that $p \in T_0$ and $p \neq a$. Then, $p \in H_0$, $\langle \mathbf{H}, \mathbf{T}\rangle, 0 \models H(r)$ and $\langle \mathbf{H}, \mathbf{T}\rangle, 0 \models r$.

– Case $a \in H(r)$: We know that $\langle \mathbf{T}, \mathbf{T} \rangle, 0 \not\models B(r) \wedge \bigwedge_{p \in H(r) \setminus \{a\}} \neg p$. Since, by assumption, $\langle \mathbf{T}, \mathbf{T} \rangle, 0 \models B(r)$, it follows $\langle \mathbf{T}, \mathbf{T} \rangle, 0 \not\models \bigwedge_{p \in H(r) \setminus \{a\}} \neg p$ Therefore, there is $p \in H(r) \setminus \{a\}$ such that $p \in T_0$. Then $p \in H_0$, $\langle \mathbf{H}, \mathbf{T} \rangle, 0 \models H(r)$ and $\langle \mathbf{H}, \mathbf{T} \rangle, 0 \models r$.

As r is chosen arbitrarily, $\langle \mathbf{H}, \mathbf{T} \rangle \models I(P)$.

b) $\langle \mathbf{H}, \mathbf{T} \rangle, 0 \models D(P)$: $\langle \mathbf{T}, \mathbf{T} \rangle \models D(P)$, then $\langle \mathbf{T}, \mathbf{T} \rangle, i \models B(r) \rightarrow H(r)$ for all $r \in D(P)$ and $i \in [1..\lambda)$. Then, for any $r \in D(P)$ and $i \in [1..\lambda)$, $\langle \mathbf{T}, \mathbf{T} \rangle, i \not\models B(r)$ or $\langle \mathbf{T}, \mathbf{T} \rangle, i \models H(r)$. If $\langle \mathbf{T}, \mathbf{T} \rangle, i \not\models B(r)$ then, by persistence, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$, and $\langle \mathbf{H}, \mathbf{T} \rangle, i \models r$. If, $\langle \mathbf{T}, \mathbf{T} \rangle, i \models H(r)$, there is some $p \in H(r)$ such that $p \in T_i$. $H_i = T_i$, so $p \in H_i$ and $\langle \mathbf{H}, \mathbf{T} \rangle, i \models H(r)$. Then $\langle \mathbf{H}, \mathbf{T} \rangle, i \models r$. As r and i are chosen arbitrarily, $\langle \mathbf{H}, \mathbf{T} \rangle \models D(P)$.

c) $\langle \mathbf{H}, \mathbf{T} \rangle, 0 \models F(P)$: final rules are constraints, so $\langle \mathbf{T}, \mathbf{T} \rangle \models F(P)$ implies $\langle \mathbf{H}, \mathbf{T} \rangle \models F(P)$.

We showed that $\langle \mathbf{H}, \mathbf{T} \rangle, 0 \models P$: a contradiction.

- If $i > 0$: we follow a very similar reasoning as for the case $i = 0$.

2. $\mathbf{T}, i \not\models a$ but $\mathbf{T}, i \models \bigvee_{r \in I(P), a \in H(r)} (\mathbf{I} \wedge S(r, a)) \vee \bigvee_{r \in D(P), a \in H(r)} (\neg \mathbf{I} \wedge S(r, a))$: again, we consider two cases here
   - there exists $r \in I(P), a \in H(r)$ and $\mathbf{T}, i \models \mathbf{I} \wedge S(r, a)$: in this case, it follows that $i = 0$, so $\mathbf{T}, 0 \models a$ and $\mathbf{T}, 0 \models S(r, a)$. Therefore, $\mathbf{T}, 0 \models B(r)$. Since $\mathbf{T}, 0 \models r$ and $\mathbf{T}, 0 \models B(r)$ then $\mathbf{T}, 0 \models p$ for some $p \in H(r)$, which contradicts $\mathbf{T}, 0 \not\models a$ and $\mathbf{T}, 0 \models \neg q$ for all $p \in H(r) \setminus \{a\}$.
   - there exists $r \in D(P), a \in H(r)$ and $\mathbf{T}, i \models \neg \mathbf{I} \wedge S(r, a)$: in this case we conclude that $i > 0$ and so $\mathbf{T}, i \models a$ and $\mathbf{T}, i \models S(r, a)$. Therefore, $\mathbf{T}, i \models B(r)$. Since $\mathbf{T}, 0 \models r$ and $\mathbf{T}, i \models B(r)$ then $\mathbf{T}, i \models q$ for some $q \in H(r)$. However, from $\mathbf{T}, i \models S(r, a)$ and $\mathbf{T}, i \not\models a$ we conclude that $\mathbf{T}, i \not\models p$ for all $p \in H(r)$: a contradiction.

For the converse direction, assume, again, by contradiction that $\langle \mathbf{T}, \mathbf{T} \rangle$ is not a TEL$_f$ model of P. We consider two cases:

1. $\langle \mathbf{T}, \mathbf{T} \rangle, 0 \not\models P$. Therefore, there exists $r \in P$ such that $\langle \mathbf{T}, \mathbf{T} \rangle, 0 \not\models r$. Clearly, r cannot be a constraint, otherwise we would already reach a contradiction. We still have to check two cases:
   - If $r \in I(P)$, then $\langle \mathbf{T}, \mathbf{T} \rangle, 0 \models B(r)$ and $\langle \mathbf{T}, \mathbf{T} \rangle, 0 \not\models H(r)$. Take any $a \in H(r)$. It follows that $\langle \mathbf{T}, \mathbf{T} \rangle, 0 \models \mathbf{I} \wedge S(r, a)$. so $\langle \mathbf{T}, \mathbf{T} \rangle, 0 \not\models CF_P(a)$: a contradiction.
   - If $r \in D(P)$ we follow a similar reasoning as for the previous case.
2. $\langle \mathbf{T}, \mathbf{T} \rangle, 0 \models P$ but $\langle \mathbf{H}, \mathbf{T} \rangle, 0 \models P$ for some $\mathbf{H} < \mathbf{T}$. By definition, there exists $i \geq 0$ such that $H_i \subset T_i$. Let us take the smallest i satisfying this property. Therefore, $H_j = T_j$ for all $j \in [0..i)$. .Moreover, Let us take $a \in T_i \setminus H_i$ and let us proceed depending on the value of i
   - If $i > 0$ and $a \in T_i$ then $\langle \mathbf{T}, \mathbf{T} \rangle, i \models CF_P(a)$ then there exists $r \in D(P)$ such that $\langle \mathbf{T}, \mathbf{T} \rangle, i \models S(r, a)$. Therefore $H(t) \setminus \{a\} \cup T_i = 0$ Since $a \notin H_i$ then $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models H(r)$ and $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$.

At this point of the proof we have that $\langle \mathbf{T}, \mathbf{T} \rangle, i \models B(r)$, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$ and $T_j \setminus H_j = \emptyset$ for any $j < i$. By Lemma 1, there must be some $b \in T_i \setminus H_i$ with a present and positive occurence in $B(r)$ that is not in the scope of negation. Then, for any $a \in T_i \setminus H_i$, there is some $b \in T_i \setminus H_i$ such that $(a, b) \in G(D(P))$. P is tight, so $G(D(P))$ is acyclic. Then, there is a topological ordering of the nodes in $G(D(P))$, and therefore of the atoms in $T_i \setminus H_i$, such that if $a, b \in T_i \setminus H_i$ and $(a, b) \in G(D(P))$, then a appears before b in the topoligical ordering. Then, there is no outgoing edge from the last node in the ordering, which contradict the fact that, for any $a \in T_i \setminus H_i$, there is some $b \in T_i \setminus H_i$ such that $(a, b) \in G(D(P))$.

- If $i = 0$ we proceed as in the previous case.

□

Proof of Theorem 2.

We first prove that if $\mathbf{T}$ is a temporal answer set of P, then $\mathbf{T}$ is a LTL$_f$-model of $CF(P)$ and $LF(P)$. The proof for $CF(P)$ is the same as for Theorem 1. Remains to prove that $\mathbf{T}$ is a LTL$_f$-model of $LF(P)$. Assume by contradiction that $\langle \mathbf{T}, \mathbf{T} \rangle \not\models LF(P)$. Two different cases must be considered:

- there is a loop L in $G(D(P))$ such that $\langle \mathbf{T}, \mathbf{T} \rangle, i \not\models \bigvee_{a \in L} a \rightarrow ES_{D(P)}(L)$ for some $i \in [1..\lambda)$, or
- there is a loop L in $G(I(P))$ such that $\langle \mathbf{T}, \mathbf{T} \rangle, 0 \not\models \bigvee_{a \in L} a \rightarrow ES_{I(P)}(L)$.

For the first case, let $\mathbf{H}$ be a trace of length λ such that $H_i = T_i \setminus L$ and $H_k = T_k$ otherwise. We show that $\langle \mathbf{H}, \mathbf{T} \rangle, 0 \models P$, which will contradict the hypothesis $\mathbf{T}$ is a TEL$_f$-model of P:

1. $\langle \mathbf{H}, \mathbf{T} \rangle, 0 \models I(P)$: follows from $\langle \mathbf{T}, \mathbf{T} \rangle \models I(P)$ by Lemma 1 as $T_0 \setminus H_0 = \emptyset$.
2. $\langle \mathbf{H}, \mathbf{T} \rangle, 0 \models F(P)$: follows from $\langle \mathbf{T}, \mathbf{T} \rangle \models F(P)$ as rules in $F(P)$ are constraints.
3. $\langle \mathbf{H}, \mathbf{T} \rangle, 0 \models D(P)$: $\langle \mathbf{T}, \mathbf{T} \rangle, 0 \models D(P)$ since $\mathbf{T}$ is a TEL$_f$-model of P. Therefore, $\langle \mathbf{T}, \mathbf{T} \rangle, k \models r$ for all $k \in [1..\lambda)$ and for all $r \in D(P)$. Then, $\langle \mathbf{T}, \mathbf{T} \rangle, k \not\models B(r)$ or $\langle \mathbf{T}, \mathbf{T} \rangle, k \models H(r)$. If $\langle \mathbf{T}, \mathbf{T} \rangle, k \not\models B(r)$, by persistence, $\langle \mathbf{H}, \mathbf{T} \rangle, k \not\models B(r)$ and $\langle \mathbf{H}, \mathbf{T} \rangle, k \models r$. If $\langle \mathbf{T}, \mathbf{T} \rangle, k \models B(r)$, then $\langle \mathbf{T}, \mathbf{T} \rangle, k \models H(r)$.
   In the case $k \neq i$, $H_k = T_k$ and $\langle \mathbf{T}, \mathbf{T} \rangle, k \models H(r)$ imply $\langle \mathbf{H}, \mathbf{T} \rangle, k \models H(r)$.In the case $k = i$, we have two cases.
   - if $\langle \mathbf{T}, \mathbf{T} \rangle, i \not\models S_{B(r)}(L)$, then, as $(T_i \setminus H_i) \setminus L = \emptyset$ and $T_k \setminus H_k = \emptyset$ for $k < i$, by Lemma 2, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$. So $\langle \mathbf{H}, \mathbf{T} \rangle, i \models r$.
   - if $\langle \mathbf{T}, \mathbf{T} \rangle, i \models S_{B(r)}(L)$ and $H(r) \cap L = \emptyset$ then $\langle \mathbf{H}, \mathbf{T} \rangle, i \models H(r)$ follows from $\langle \mathbf{T}, \mathbf{T} \rangle, i \models H(r)$ and $\langle \mathbf{H}, \mathbf{T} \rangle, i \models r$.
   - if $\langle \mathbf{T}, \mathbf{T} \rangle, i \models S_{B(r)}(L)$ and $H(r) \cap L = \emptyset$ then
     - if there is some $p \in H(r) \setminus L$ such that $p \in T_i$, then $p \in H_i$. So $\langle \mathbf{H}, \mathbf{T} \rangle, i \models H(r)$ and then $\langle \mathbf{H}, \mathbf{T} \rangle, i \models r$.
     - if there is no $p \in H(r) \setminus L$ such that $p \in T_i$, then $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus L} \neg p$. As we also have $\langle \mathbf{T}, \mathbf{T} \rangle, i \models S_{B(r)}(L)$, $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigvee_{a \in L} a \rightarrow ES_{D(P)}(L)$, which contradict our hypothesis.

The proof of the second case follows a similar reasoning as for the first one.

Next, we prove that if $\mathbf{T}$ is a LTL$_f$-model of $CF(P)$ and $LF(P)$, then $\mathbf{T}$ is a TEL$_f$-model of P. The proof for $\langle \mathbf{T}, \mathbf{T} \rangle \models P$ is the same as for Theorem 1. Remains to prove that there is no $\mathbf{H} < \mathbf{T}$ such that $\langle \mathbf{H}, \mathbf{T} \rangle \models P$.

Let assume that there exists such a trace $\mathbf{H}$, and let i be the smallest time point such that $H_i \subset T_i$. Therefore, $H_k = T_k$ for all $k \in [0..i)$.

- If $i > 0$: Let $a \in T_i \setminus H_i$. $\langle \mathbf{T}, \mathbf{T} \rangle \models CF(P)$, so $\langle \mathbf{T}, \mathbf{T} \rangle, i \models a \leftrightarrow \bigvee_{r \in D(P), a \in H(r)} (B(r) \wedge \bigwedge_{p \in H(r) \setminus \{a\}} \neg p)$. As $a \in T_i$, there is some rule $r \in D(P)$ such that $a \in H(r)$, $\langle \mathbf{T}, \mathbf{T} \rangle, i \models B(r)$, and $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus \{a\}} \neg p$. $\langle \mathbf{H}, \mathbf{T} \rangle \models P$, so $\langle \mathbf{H}, \mathbf{T} \rangle, i \models B(r) \rightarrow a \vee \bigvee_{p \in H(r) \setminus \{a\}} p$. Then, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$ or $\langle \mathbf{H}, \mathbf{T} \rangle, i \models a$ or $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \bigvee_{p \in H(r) \setminus \{a\}} p$. As $a \notin H_i$, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models a$. As $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus \{a\}} \neg p$, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models \bigvee_{p \in H(r) \setminus \{a\}} p$. So, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$.
  $\langle \mathbf{T}, \mathbf{T} \rangle, i \models B(r)$, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$ and $T_j \setminus H_j = \emptyset$ for $j < i$, so, by Lemma 1, there must be some $b \in T_i \setminus H_i$ with a present and positive occurence in $B(r)$ that is not in the scope of negation. Therefore, for any $a \in T_i \setminus H_i$, there is some $b \in T_i \setminus H_i$ such that $(a, b) \in G(D(P))$. It implies a loop L in $D(P)$, with $L \subseteq T_i \setminus H_i$.
  The strongly connected components (SCC) of the dependency graph of $D(P)$ over $T_i \setminus H_i$ form a directed acyclic graph, so there is some SCC L, such that, for any $a \in L$, there is no $b \in (T_i \setminus H_i) \setminus L$ such that $(a, b) \in G(D(P))$.
  For any $a \in T_i \setminus H_i$, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$, for all $r \in D(P)$ such that $a \in H(r)$ and $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus \{a\}} \neg p$. So $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$, for all $r \in D(P)$ such that $L \cap H(r) \neq \emptyset$ and $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus L} \neg p$. Let $\mathbf{X}$ be a trace of length λ with $X_i = L$ and $X_j = \emptyset$ for $j \neq i$. For any $a \in L$ there is no $b \in (T_i \setminus H_i) \setminus L$ such that $(a, b) \in G(D(P))$, so all positive and present occurences of atoms from L in $B(r)$ are in the scope of negation. Then, we can apply Lemma 1, and get that $\langle \mathbf{T} \setminus \mathbf{X}, \mathbf{T} \rangle, i \not\models B(r)$, for all $r \in D(P)$ such that $L \cap H(r) \neq \emptyset$ and $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus L} \neg p$. Then, as $X_i \setminus L = \emptyset$, by Lemma 2, $\langle \mathbf{T}, \mathbf{T} \rangle, i \not\models S_{B(r)}(L)$, for all $r \in D(P)$ such that $L \cap H(r) \neq \emptyset$ and $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus L} \neg p$. So, $\langle \mathbf{T}, \mathbf{T} \rangle, i \not\models \bigvee_{a \in L} a \rightarrow ES_{D(P)}(L)$, and then $\langle \mathbf{T}, \mathbf{T} \rangle \not\models LF(P)$. Contradiction.

- Case $i = 0$: we reach a contradiction in a similar way as above.

□

Proof of Theorem 3.

We first prove that if $\mathbf{T}$ is a temporal answer set of P, then $\mathbf{T}$ is a LTL$_f$-model of $P \cup LF(P)$. $\mathbf{T}$ is a temporal answer set of P, so $\mathbf{T}$ is a LTL$_f$-model of P. We can show that $\mathbf{T}$ is a LTL$_f$-model of $LF(P)$ the same way as for Theorem 2.

Next, we prove that if $\mathbf{T}$ is a LTL$_f$-model of $P \cup LF(P)$, then $\mathbf{T}$ is a temporal answer set of P. It amounts to showing that there is no $\mathbf{H} < \mathbf{T}$ such that $\langle \mathbf{H}, \mathbf{T} \rangle \models P$. Let assume that there is such a trace $\mathbf{H}$, and let i be the smallest time point such that $H_i \subset T_i$.

1. If $i > 0$, Let $a \in T_i \setminus H_i$. $\langle \mathbf{T}, \mathbf{T} \rangle \models LF(P)$, so $\langle \mathbf{T}, \mathbf{T} \rangle, i \models a \leftrightarrow \bigvee_{r \in D(P), a \in H(r)} (S_{B(r)}(a) \wedge \bigwedge_{p \in H(r) \setminus \{a\}} \neg p)$. As $a \in T_i$, there exists $r \in D(P)$ such that $a \in H(r)$, $\langle \mathbf{T}, \mathbf{T} \rangle, i \models$

$S_{B(r)}(a)$ and $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus \{a\}} \neg p$. $\langle \mathbf{H}, \mathbf{T} \rangle \models P$, so $\langle \mathbf{H}, \mathbf{T} \rangle, i \models B(r) \rightarrow a \vee \bigvee_{p \in H(r) \setminus \{a\}} p$. Then, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$ or $\langle \mathbf{H}, \mathbf{T} \rangle, i \models a$ or $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \bigvee_{p \in H(r) \setminus \{a\}} p$. As $a \notin H_i$, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models a$. As $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus \{a\}} \neg p$, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models \bigvee_{p \in H(r) \setminus \{a\}} p$. So, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$.
$\langle \mathbf{T}, \mathbf{T} \rangle, i \models S_{B(r)}(a)$, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$ and $T_j \setminus H_j = \emptyset$ for $j < i$, so, by Lemma 2, there must be some $b \in T_i \setminus H_i$ with a present and positive occurence in $B(r)$ that is not in the scope of negation. Therefore, for any $a \in T_i \setminus H_i$, there is some $b \in T_i \setminus H_i$ such that $(a, b) \in G(D(P))$. It implies a loop L in $D(P)$, with $L \subseteq T_i \setminus H_i$.
The strongly connected components (SCC) of the dependency graph of $D(P)$ over $T_i \setminus H_i$ form a directed acyclic graph, so there is some SCC L, such that, for any $a \in L$, there is no $b \in (T_i \setminus H_i) \setminus L$ such that $(a, b) \in G(D(P))$.
For any $a \in T_i \setminus H_i$, $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$, for all $r \in D(P)$ such that $a \in H(r)$ and $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus \{a\}} \neg p$. So $\langle \mathbf{H}, \mathbf{T} \rangle, i \not\models B(r)$, for all $r \in D(P)$ such that $L \cap H(r) \neq \emptyset$ and $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus L} \neg p$. Let $\mathbf{X}$ be a trace of length λ with $X_i = L$ and $X_j = \emptyset$ for $j \neq i$. For any $a \in L$ there is no $b \in (T_i \setminus H_i) \setminus L$ such that $(a, b) \in G(D(P))$, so all positive and present occurences of atoms from L in $B(r)$ are in the scope of negation. Then, we can apply Lemma 1, and get that $\langle \mathbf{T} \setminus \mathbf{X}, \mathbf{T} \rangle, i \not\models B(r)$, for all $r \in D(P)$ such that $L \cap H(r) \neq \emptyset$ and $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus L} \neg p$. Then, as $X_i \setminus L = \emptyset$, by Lemma 2, $\langle \mathbf{T}, \mathbf{T} \rangle, i \not\models S_{B(r)}(L)$, for all $r \in D(P)$ such that $L \cap H(r) \neq \emptyset$ and $\langle \mathbf{T}, \mathbf{T} \rangle, i \models \bigwedge_{p \in H(r) \setminus L} \neg p$. So, $\langle \mathbf{T}, \mathbf{T} \rangle, i \not\models \bigvee_{a \in L} a \rightarrow ES_{D(P)}(L)$, and then $\langle \mathbf{T}, \mathbf{T} \rangle \not\models LF(P)$: a contradiction.

2. For the case when $i = 0$ we reach a contradiction in a similar way as above.

$\square$