

*Towards Constraint Temporal Answer Set Programming**

PEDRO CABALAR

University of Corunna, Corunna, Spain
(e-mail: cabalar@udc.es)

MARTÍN DIÉGUEZ

University of Angers, Angers, France
(e-mail: martin.dieuezlodeiro@univ-angers.fr)

FRANÇOIS OLIVIER

CRIL CNRS & Artois University, France
(e-mail: olivier@cril.fr)

TORSTEN SCHAUB

University of Potsdam, Potsdam, Germany
Potassco Solutions, Potsdam, Germany
(e-mail: torsten@cs.uni-potsdam.de)

IGOR STÉPHAN

University of Angers, Angers, France
(e-mail: igor.stephan@univ-angers.fr)

submitted 24 July 2025; revised 24 July 2025; accepted 27 July 2025

Abstract

Reasoning about dynamic systems with a fine-grained temporal and numeric resolution presents significant challenges for logic-based approaches like Answer Set Programming (ASP). To address this, we introduce and elaborate upon a novel temporal and constraint-based extension of the logic of Here-and-There and its nonmonotonic equilibrium extension, representing, to the best of our knowledge, the first approach to nonmonotonic temporal reasoning with constraints specifically tailored for ASP. This expressive system is achieved by a synergistic combination of two foundational ASP extensions: the linear-time logic of Here-and-There, providing robust non-monotonic temporal reasoning capabilities, and the logic of Here-and-There with constraints, enabling the direct integration and manipulation of numeric constraints, among others. This work establishes the foundational logical framework for tackling complex dynamic systems with high resolution within the ASP paradigm.

KEYWORDS: temporal logic programming, nonmonotonic reasoning, knowledge representation, constraint programming

* An extended version of this paper can be found at <https://arxiv.org/abs/2507.13958>

1 Introduction

Reasoning about action and change is crucial for understanding how dynamic systems evolve over time and how actions influence those changes. Representing dynamic systems with higher resolution, such as by employing finer time units and more precise numeric variables, significantly increases the complexity of reasoning about them, posing particular challenges for logic-based approaches like Answer Set Programming (ASP; Lifschitz 2008). To illustrate this, consider the following scenario: “A radar is positioned at the 400 km mark on a road with a speed limit of 90 km/h. A car is initially traveling at 80 km/h. At time instant 4, the car accelerates by 11.35 km/h. Subsequently, at time instant 6, it decelerates by 2.301 km/h. The problem is to determine whether the car will exceed the speed limit and thus incur a fine.” A closer look reveals several key numeric entities: the car’s position and speed, the radar’s position and speed limit, and the car’s acceleration and deceleration.

Our objective is to provide the logical foundations for extending ASP to effectively model such scenarios involving changing numeric values over time. To achieve this, we integrate two key logical formalisms that extend ASP: the linear-time logic of Here-and-There (THT; Aguado et al. 2023), with its ability to handle nonmonotonic temporal reasoning, and the logic of Here-and-There with constraints (HT_c; Cabalar et al. 2016), which allows us to incorporate and reason about numeric constraints, among others. Building upon standard linear temporal logic, THT and its nonmonotonic equilibrium extension, TEL, enable the expression of sophisticated temporal behaviors in dynamic systems through features like inertia and default reasoning. HT_c complements this by allowing us to directly reason with numeric and other constraints. Furthermore, its equilibrium extension provides solid logical foundations for tackling nonmonotonic constraint satisfaction problems, elegantly handling situations with incomplete information by using default values. With this motivation established, we now proceed to introduce the combined logical framework.

2 Temporal here-and-there with constraints

The syntax of the logic *Temporal Here-and-There with constraints* (THT_c) relies on a signature $\langle \mathcal{X}, \mathcal{D}, \mathcal{A} \rangle$, akin to constraint satisfaction problems (CSPs; Dechter 2003). Specifically, $\mathcal{X}$ denotes a set of variables and $\mathcal{D}$ represents the domain of values, often identified with their corresponding constants. The set $\mathcal{A}$ comprises *temporal constraint atoms* (or simply *atoms*), which are defined over temporal terms.

A *temporal term* (or simply *term*) is fundamental to THT_c and is intended to represent the value of a variable at past, present, or future time points. We represent such a term by an expression $\circ^i x$ where $x \in \mathcal{X}$ and $i \in \mathbb{Z}$, while overloading the temporal modal operator for “next”, viz. $\circ$. The integer i indicates a temporal offset: a positive i signifies i steps forward in time to retrieve the value of x , a negative i signifies $|i|$ steps backward, and $i = 0$ refers to the value of x in the current state. For notational convenience, we use x and $\circ x$ as shorthand for $\circ^0 x$ and $\circ^1 x$, respectively. Furthermore, we overload the operator $\bullet$ for “previous” to represent past variable values. That is, $\bullet^i x$ stands for $\circ^{-i} x$ for offset i . For example, the constraint $\circ x + \bullet y \leq z$ is equivalent to $\circ^1 x + \circ^{-1} y \leq \circ^0 z$.

Accordingly, an n -ary temporal constraint atom $c \in \mathcal{A}$ is often represented as $c(\circ^{o_1}x_1, \dots, \circ^{o_n}x_n)$ where $\circ^{o_1}x_1, \dots, \circ^{o_n}x_n$ are (possibly identical) temporal terms. To further illustrate, the atom $(\bullet^3x = 4)$ can be read as “ x had value 4 three states ago”, and $(\circ x = x)$ as “the value of x in the next state is identical to its current value”.

Then, *temporal constraint formulas* (or just *formulas*) are defined as follows:

$$\varphi ::= c \in \mathcal{A} \mid \perp \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \circ \varphi \mid \varphi \mathbb{U} \varphi \mid \varphi \mathbb{R} \varphi \mid \bullet \varphi \mid \varphi \mathbf{S} \varphi \mid \varphi \mathbf{T} \varphi$$

Connectives $\perp, \wedge, \vee$ and $\rightarrow$ are *Boolean*, while the remaining connectives are *temporal* modalities. We use outlined operators to refer to future modalities and solid ones for past modalities. Accordingly, $\circ, \mathbb{U}, \mathbb{R}$ represent the future modalities *next*, *until*, *release*, respectively, while $\bullet, \mathbf{S}, \mathbf{T}$ reflect their past counterparts: *previous*, *since*, *trigger*.

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

$$\begin{array}{llll} \mathbf{I} \stackrel{\text{def}}{=} \neg \bullet \top & \hat{\bullet} \varphi \stackrel{\text{def}}{=} \bullet \varphi \vee \mathbf{I} & \blacksquare \varphi \stackrel{\text{def}}{=} \perp \mathbf{T} \varphi & \blacklozenge \varphi \stackrel{\text{def}}{=} \top \mathbf{S} \varphi \\ \mathbb{F} \stackrel{\text{def}}{=} \neg \circ \top & \hat{\circ} \varphi \stackrel{\text{def}}{=} \circ \varphi \vee \mathbb{F} & \square \varphi \stackrel{\text{def}}{=} \perp \mathbb{R} \varphi & \diamond \varphi \stackrel{\text{def}}{=} \top \mathbb{U} \varphi \end{array}$$

Specifically, $\mathbf{I}$ and $\mathbb{F}$ refer to the “initial” and “final” state, respectively; $\hat{\bullet} \varphi$ ($\hat{\circ} \varphi$) refers to φ at the previous (next) state in case that it exists; $\blacksquare \varphi$ ($\square \varphi$) is read as “ φ has always been true” (“ φ will always be true”), and $\blacklozenge \varphi$ ($\diamond \varphi$) is read as “ φ has been true” (“ φ will eventually be true”). For instance, $\square(\circ x > x)$ allows us to express informally that “the value of variable x is increasing over time”.

We define the semantics of HT_c in terms of HT_c traces, which are sequences of HT_c interpretations (Cabalar *et al.* 2016). We first present the necessary semantic concepts of HT_c before further elaborating on these traces. HT_c relies on partial valuations, that is, functions $v: \mathcal{X} \rightarrow \mathcal{D} \cup \{u\}$, where the range is augmented by the special value $u \notin \mathcal{D}$ representing “undefined”. We define $\mathcal{D}_u \stackrel{\text{def}}{=} \mathcal{D} \cup \{u\}$ and denote such valuations also by $v: \mathcal{X} \rightarrow \mathcal{D}_u$. Given two partial valuations v and v' , we define $v \sqsubseteq v'$ if $v(x) = d$ implies $v'(x) = d$ for all $x \in \mathcal{X}$ and $d \in \mathcal{D}$. Similarly, we say that $v = v'$ if $v \sqsubseteq v'$ and $v' \sqsubseteq v$, and $v \sqsubset v'$ if $v \sqsubseteq v'$ and $v \neq v'$. An HT_c interpretation is a pair $\langle v_h, v_t \rangle$ of valuations such that $v_h \sqsubseteq v_t$.

Building upon this, we define a *trace* $\mathbf{v}$ of length λ as a sequence $\mathbf{v} = (v_i)_{i \in [0..\lambda)}$ of partial valuations v_i for $i \in [0..\lambda)$. For instance, the sequence $\mathbf{v}' = \{x \mapsto 4\} \cdot \{x \mapsto u\} \cdot \{x \mapsto 5\}$ represents a finite trace with three valuations (separated by “.”) that assign different values to x except at the second state where x is undefined. The $\sqsubseteq$ relation can be extended to cope with traces in the following way: given two traces $\mathbf{v}_h = (v_{h,i})_{i \in [0..\lambda)}$ and $\mathbf{v}_t = (v_{t,i})_{i \in [0..\lambda)}$ of length λ , we define $\mathbf{v}_h \sqsubseteq \mathbf{v}_t$ if $v_{h,i} \sqsubseteq v_{t,i}$ for all $i \in [0..\lambda)$. For instance, our previous example trace $\mathbf{v}'$ satisfies $\mathbf{v}' \sqsubseteq \{x \mapsto 4\} \cdot \{x \mapsto 1\} \cdot \{x \mapsto 5\}$ because both traces coincide in the variables defined in $\mathbf{v}'$, but the new trace assigns $x \mapsto 1$ at the second state, while $\mathbf{v}'$ left it undefined. As before, we say $\mathbf{v}_h = \mathbf{v}_t$ if $v_{h,i} = v_{t,i}$ for all $i \in [0..\lambda)$ and $\mathbf{v}_h \sqsubset \mathbf{v}_t$ if $\mathbf{v}_h \sqsubseteq \mathbf{v}_t$ and $\mathbf{v}_h \neq \mathbf{v}_t$. All this allows us to define an $h\text{HT}_c$ trace of length λ as a pair $\langle \mathbf{v}_h, \mathbf{v}_t \rangle$ of traces $\mathbf{v}_h$ and $\mathbf{v}_t$ of length λ such that $\mathbf{v}_h \sqsubseteq \mathbf{v}_t$.

HT_c evaluates constraint atoms using denotations, a mechanism that abstracts from the syntax and semantics of expressions originating from external theories. In HT_c , a valuation can be seen as a solution of a constraint. For instance the denotation $\llbracket x + y \leq 5 \rrbracket$ contains all valuations v such that $v(x) + v(y) \leq 5$. Also, the satisfaction of a constraint atom corresponds in HT_c to checking whether a valuation belongs to a denotation or not. In THT_c , however, time-varying variables are handled. For instance the constraint $x + \circ^5 x \leq 5$ is satisfied at time point t when the addition of the values of x at t and $t + 2$ is smaller or equal to 5. In the temporal case, evaluating a temporal constraint atom requires the use of a finite number of valuations, for which we would need a more complex denotational approach. Therefore, we adopt a simpler approach, associating each constraint atom with a relation, where each tuple signifies a valid value assignment, as is standard in CSPs: Given a temporal constraint atom c of arity n , we define its *solution relation* as $[c] \subseteq \mathcal{D}_u^n$. Note that this definition tolerates undefined variables. Given a constraint atom c of arity n , $[c]$ is said to be *strict* whenever $[c] \subseteq \mathcal{D}^n$, that is, if it leaves no variables undefined. Otherwise, we say that $[c]$ is *non-strict*.

For instance, to satisfy the atom $x + \circ^5 x \leq 5$ neither term x nor $\circ^5 x$ can be undefined. A natural choice for its (strict) solution relation is thus

$$[x + \circ^5 x \leq 5] = \{(a, b) \in \mathbb{Z}^2 \mid a + b \leq 5\}.$$

As an example of an atom with a non-strict solution, we could define an atom such as *some_zero*($\bullet x, x, \circ x$) requiring that x equals zero at the previous, current, or next state, but allowing for x to be undefined in some of those states:

$$[\text{some_zero}(\bullet x, x, \circ x)] = \{(a, b, c) \in (\mathbb{Z} \cup \{u\})^3 \mid a = 0 \text{ or } b = 0 \text{ or } c = 0\}.$$

Given a trace $\mathbf{v} = (v_i)_{i \in [0..\lambda]}$ of length λ , the *value* of a term $\circ^o x$ at time point $i \in [0..\lambda]$ is defined as

$$v_i(\circ^o x) \stackrel{\text{def}}{=} \begin{cases} v_{i+o}(x) & \text{if } i + o \in [0..\lambda] \\ u & \text{otherwise.} \end{cases}$$

Given an HT_c trace $\mathbf{M} = \langle \mathbf{v}_h, \mathbf{v}_t \rangle$ and a time point $i \in [0..\lambda]$, we define the satisfaction of a formula in THT_c as follows:

1. $\mathbf{M}, i \models c(\circ^{o_1} x_1, \dots, \circ^{o_n} x_n)$ if for all $w \in \{h, t\}$ $(v_{w,i}(\circ^{o_1} x_1), \dots, v_{w,i}(\circ^{o_n} x_n)) \in [c(\circ^{o_1} x_1, \dots, \circ^{o_n} x_n)]$
2. $\mathbf{M}, i \models \varphi \wedge \psi$ if $\mathbf{M}, i \models \varphi$ and $\mathbf{M}, i \models \psi$
3. $\mathbf{M}, i \models \varphi \vee \psi$ if $\mathbf{M}, i \models \varphi$ or $\mathbf{M}, i \models \psi$
4. $\mathbf{M}, i \models \varphi \rightarrow \psi$ if $\langle \mathbf{v}_w, \mathbf{v}_t \rangle, i \not\models \varphi$ or $\langle \mathbf{v}_w, \mathbf{v}_t \rangle, i \models \psi$ for all $w \in \{h, t\}$
5. $\mathbf{M}, i \models \circ \varphi$ if $i < \lambda - 1$ and $\mathbf{M}, i + 1 \models \varphi$
6. $\mathbf{M}, i \models \varphi \mathbb{U} \psi$ if there exists $k \in [i..\lambda]$ s.t. $\mathbf{M}, k \models \psi$ and $\mathbf{M}, j \models \varphi$ for all $i \leq j < k$
7. $\mathbf{M}, i \models \varphi \mathbb{R} \psi$ if for all $k \in [i..\lambda]$, either $\mathbf{M}, k \models \psi$ or $\mathbf{M}, j \models \varphi$ for some $i \leq j < k$
8. $\mathbf{M}, i \models \bullet \varphi$ if $i > 0$ and $\mathbf{M}, i - 1 \models \varphi$
9. $\mathbf{M}, i \models \varphi \mathbf{S} \psi$ if there exists $k \in [0..i]$ s.t. $\mathbf{M}, k \models \psi$ and $\mathbf{M}, j \models \varphi$ for all $k < j \leq i$
10. $\mathbf{M}, i \models \varphi \mathbf{T} \psi$ if for all $k \in [0..i]$, either $\mathbf{M}, k \models \psi$ or $\mathbf{M}, j \models \varphi$ for some $k < j \leq i$

In fact, given a constraint atom c with a strict relation $[c]$, the satisfaction of c depends solely on the “here” component of the trace (Cabalar *et al.* 2016). In this case, Condition 1 can be replaced by:

$$1' \quad \mathbf{M}, i \models c(\circ^{o_1}x_1, \dots, \circ^{o_n}x_n) \text{ if } (v_{h,i}(\circ^{o_1}x_1), \dots, v_{h,i}(\circ^{o_n}x_n)) \in [c(\circ^{o_1}x_1, \dots, \circ^{o_n}x_n)].$$

Proposition 1.

Given an HT_c trace $\mathbf{M} = \langle \mathbf{v}_h, \mathbf{v}_t \rangle$ and a time point $i \in [0.. \lambda)$, we have the following satisfaction relations for the derived operators:

1. $\mathbf{M}, i \models \blacksquare\varphi$ if for all $j \in (0..i]$, $\mathbf{M}, j \models \varphi$
2. $\mathbf{M}, i \models \blacklozenge\varphi$ if there exists $j \in (0..i]$ such that $\mathbf{M}, j \models \varphi$
3. $\mathbf{M}, i \models \mathbf{I}$ if $i = 0$
4. $\mathbf{M}, i \models \widehat{\circ}\varphi$ if either $i = 0$ or $\mathbf{M}, i - 1 \models \varphi$
5. $\mathbf{M}, i \models \square\varphi$ if for all $j \in [i.. \lambda)$, $\mathbf{M}, j \models \varphi$
6. $\mathbf{M}, i \models \Diamond\varphi$ if there exists $j \in [i.. \lambda)$ such that $\mathbf{M}, j \models \varphi$
7. $\mathbf{M}, i \models \mathbb{F}$ if $i = \lambda - 1$
8. $\mathbf{M}, i \models \widehat{\circ}\varphi$ if either $i = \lambda - 1$ or $\mathbf{M}, i + 1 \models \varphi$

For illustration, consider the short HT_c trace $\langle \mathbf{v}_h, \mathbf{v}_t \rangle$ where

$$\begin{aligned} \mathbf{v}_h &= \{x \mapsto 4, y \mapsto u\} \cdot \{x \mapsto 5, y \mapsto u\} \cdot \{x \mapsto u, y \mapsto u\} \cdot \{x \mapsto 5, y \mapsto 6\} \\ \mathbf{v}_t &= \{x \mapsto 4, y \mapsto 6\} \cdot \{x \mapsto 5, y \mapsto u\} \cdot \{x \mapsto 4, y \mapsto 5\} \cdot \{x \mapsto 5, y \mapsto 6\} \end{aligned}$$

As a first example, consider the formula $(x = 4) \wedge (\circ x < \circ^3 y)$ whose atoms have the solution relations $[x = 4] = \{(4)\}$ and $[\circ x < \circ^3 y] = \{(a, b) \in \mathbb{Z}^2 \mid a < b\}$. Note that each solution relation above contains only the assignments that satisfy the constraint and they do not depend on the “next” $\circ^k$ prefix that qualifies each variable inside a constraint. Broadly speaking, a solution for $(\circ x < \circ^3 y)$ consists of two assignments, one for x and one for y that satisfy the constraint. The effect of using the term $\circ^3 y$ is that the value for y is given by the valuation placed three states ahead in the trace, that is, the valuation of $\circ^3 y$ depends on the sequence of valuations in the trace. Moreover, both $[x = 4]$ and $[\circ x < \circ^3 y]$ are strict, so we can use Condition 1' to evaluate the constraint atoms. We have therefore $\langle \mathbf{v}_h, \mathbf{v}_t \rangle, 0 \models (x = 4)$. Also, we have $\langle \mathbf{v}_h, \mathbf{v}_t \rangle, 0 \models (\circ x < \circ^3 y)$ since $v_{h,3}(y) = 6$, $v_{h,1}(x) = 5$ and $(5, 6) \in [\circ x < \circ^3 y]$. Therefore, $\langle \mathbf{v}_h, \mathbf{v}_t \rangle, 0 \models (x = 4) \wedge (\circ x < \circ^3 y)$.

Next, consider the formula $(\bullet x < 7)$ along with¹ $[\bullet x < 7] = \{a \in \mathbb{Z} \mid a < 7\}$. Since there is no time point before the initial state, we get $v_{h,0}(\bullet x) = u$. Hence, $\langle \mathbf{v}_h, \mathbf{v}_t \rangle, 0 \not\models \bullet x < 7$ because $u \notin [\bullet x < 7]$. However, if the constraint is evaluated at time point 1, we get $\langle \mathbf{v}_h, \mathbf{v}_t \rangle, 1 \models \bullet x < 7$ as $v_{h,1}(\bullet x) = 4$ and $4 \in [\bullet x < 7]$.

Finally, consider the equation $y = y$ with $[y = y] = \mathbb{Z}$. Clearly, we have $\langle \mathbf{v}_h, \mathbf{v}_t \rangle, 3 \models y = y$ since $v_{h,3}(y) = 6$ and $6 \in [y = y]$. However, this is not the case at time point 0 because $v_{h,0}(y) = u$ and $u \notin [y = y]$.

Next, we show that THT_c satisfies the characteristic properties of HT-based logics.

Proposition 2

(Persistence, Negation). For all HT_c traces $\langle \mathbf{v}_h, \mathbf{v}_t \rangle$ of length λ , all formulas φ , and all $i \in [0.. \lambda)$ we have:

¹ We sometimes identify 1-tuples like $\langle a \rangle$ with the element itself a .

$\langle \mathbf{v}_h, \mathbf{v}_t \rangle, i \models \varphi$ implies $\langle \mathbf{v}_t, \mathbf{v}_t \rangle, i \models \varphi$.
 $\langle \mathbf{v}_h, \mathbf{v}_t \rangle, i \models \neg \varphi$ if $\langle \mathbf{v}_t, \mathbf{v}_t \rangle, i \not\models \varphi$.

Finally, temporal equilibrium models are defined in the traditional way.

Definition 1.

An HT_c trace $\langle \mathbf{v}_t, \mathbf{v}_t \rangle$ of length λ is a temporal equilibrium model of a formula φ if

1. $\langle \mathbf{v}_t, \mathbf{v}_t \rangle, 0 \models \varphi$ and
2. there is no HT_c trace $\langle \mathbf{v}_h, \mathbf{v}_t \rangle$ such that $\mathbf{v}_h \sqsubset \mathbf{v}_t$ and $\langle \mathbf{v}_h, \mathbf{v}_t \rangle, 0 \models \varphi$.

Given that equilibrium models are the semantic counterpart of stable models in ASP, we may thus refer to $\mathbf{v}_t$ as a stable model of φ in THT_c .

2.1 THT_c as a conservative extension of HT_c

THT_c can be viewed as a temporal extension of HT_c (Cabalar et al. 2016). We support this in what follows by showing that both coincide on non-temporal formulas.

To achieve this, we need to establish a correspondence between the semantic concepts of denotations and solution relations. Given a signature $\langle \mathcal{X}, \mathcal{D}, \mathcal{A} \rangle$, let $\mathcal{V}$ be the set of all partial valuations from $\mathcal{X}$ to $\mathcal{D}_u$. A *denotation* is a function $\llbracket \cdot \rrbracket : \mathcal{A} \rightarrow 2^{\mathcal{V}}$ assigning a set of partial valuations to each atom. In HT_c , one considers denotations with a closed range: if a partial valuation v satisfies an atom c , $v \in \llbracket c \rrbracket$, then any partial valuation v' that extends v , $v \sqsubseteq v'$, also satisfies c .

$$\llbracket x = y \rrbracket \stackrel{\text{def}}{=} \{v \in \mathcal{V} \mid v(x) = v(y) \neq u\}$$

$$\llbracket \text{some_zero}(X) \rrbracket \stackrel{\text{def}}{=} \{v \in \mathcal{V} \mid v(x) = 0 \text{ for some } x \in X\}.$$

For a (non-temporal) constraint $c(x_1, \dots, x_n)$ in $\mathcal{A}$, only referring to variable values in the current state, we define

$$\llbracket c(x_1, \dots, x_n) \rrbracket = \{(v(x_1), \dots, v(x_n)) \mid v \in \mathcal{V}\}$$

$$\llbracket c(x_1, \dots, x_n) \rrbracket = \{v \in \mathcal{V} \mid (v(x_1), \dots, v(x_n)) \in \llbracket c(x_1, \dots, x_n) \rrbracket\}.$$

Given this correspondence, we can show the following result, where $\models_{\text{HT}_c}$ and $\models_{\text{THT}_c}$ denote the satisfaction relation of HT_c (Cabalar et al. 2016) and THT_c (as given above).

Proposition 3.

For all formulas φ containing only Boolean connectives and all HT_c interpretations $\langle \mathbf{v}_h, \mathbf{v}_t \rangle$, we have $\langle \mathbf{v}_h, \mathbf{v}_t \rangle \models_{\text{HT}_c} \varphi$ if $(\langle \mathbf{v}_h, \mathbf{v}_t \rangle), 0 \models_{\text{THT}_c} \varphi$.

Since HT_c interpretations are simply HT_c traces of length 1, it follows that HT_c and THT_c coincide when restricted to the language of HT_c .

2.2 THT_c as a conservative extension of THT_f

THT_c extends THT_f (Aguado et al. 2023) by including constraints. Similar to the previous section, we confirm this relationship by showing that they coincide on temporal

formulas without non-Boolean constraints. To proceed, we first need to introduce basic concepts of THT_f .

THT_f employs the same syntax for temporal formulas as THT_c , with the key distinction that constraint atoms are replaced by Boolean atoms in $\mathcal{A}$. The semantics of THT_f relies on HT traces. An underlying (Boolean) trace of length λ is a sequence $(T_i)_{i \in [0..\lambda)}$ where $T_i \subseteq \mathcal{A}$ for all $i \in [0..\lambda)$. An HT trace $\langle \mathbf{H}, \mathbf{T} \rangle$ is a pair of traces $\mathbf{H} = (H_i)_{i \in [0..\lambda)}$ and $\mathbf{T} = (T_i)_{i \in [0..\lambda)}$ satisfying $H_i \subseteq T_i$ for all $i \in [0..\lambda)$. An HT trace $\langle \mathbf{H}, \mathbf{T} \rangle$ satisfies a Boolean atom $p \in \mathcal{A}$ at $i \in [0..\lambda)$ if $p \in H_i$. The satisfaction conditions for more complex formulas are identical to those of THT_c .

To encode THT_f within THT_c , we consider the signature $\langle \mathcal{A}, \{\mathbf{t}\}, \{p = \mathbf{t} \mid p \in \mathcal{A}\} \rangle$, where $\mathbf{t}$ is considered as true.² Accordingly, we set $[p = \mathbf{t}] = \{\mathbf{t}\}$ for all $p \in \mathcal{A}$. Then, we define the bijective mapping δ from HT traces to HT_c traces as follows.³ Given an HT trace $\langle \mathbf{H}, \mathbf{T} \rangle = \langle (H_i)_{i \in [0..\lambda)}, (T_i)_{i \in [0..\lambda)} \rangle$ of length λ , we define the corresponding HT_c trace $\delta(\langle \mathbf{H}, \mathbf{T} \rangle) = \langle \mathbf{v}_h, \mathbf{v}_t \rangle$ of length λ where for all $i \in [0..\lambda)$ and for all $p \in \mathcal{A}$, we have

$$v_{h,i}(p) = \begin{cases} \mathbf{t} & \text{if } p \in H_i \\ u & \text{otherwise} \end{cases} \quad v_{t,i}(p) = \begin{cases} \mathbf{t} & \text{if } p \in T_i \\ u & \text{otherwise} \end{cases}$$

Then, the following proposition can be easily proved via structural induction.

Proposition 4.

For any HT trace $\langle \mathbf{H}, \mathbf{T} \rangle$, we have for all temporal formulas φ and all $i \in [0..\lambda)$ that $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \varphi$ if $\delta(\langle \mathbf{H}, \mathbf{T} \rangle), i \models \varphi'$, where φ' is obtained from φ by replacing every atom p by the constraint $p = \mathbf{t}$.

Proposition 4 can be extended to the case of equilibrium models.

Proposition 5.

For any HT trace $\langle \mathbf{H}, \mathbf{T} \rangle$ and for any formula φ , $\langle \mathbf{v}_h, \mathbf{v}_t \rangle$ is a temporal equilibrium model of φ if $\delta(\langle \mathbf{H}, \mathbf{T} \rangle)$ is a temporal equilibrium model (under THT_c semantics) of φ .

3 From THT_c to quantified HT with evaluable functions

Kamp's translation (1968) is a cornerstone result that provides a fundamental link between temporal and classical logic, offering deep theoretical insights and practical implications for the study and application of temporal reasoning.

As a step towards a similar result in our non-classical context, we define a translation from THT_c into Quantified⁴ HT with decidable equality, evaluable functions (Cabalar 2011), and an order relation. We consider a first-order language with signature $\langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$, where $\mathcal{C}$ and $\mathcal{F}$ are disjoint sets of uninterpreted and evaluable function names, respectively, and $\mathcal{P}$ is a set of predicate names including equality ($=$) and a strict order relation $<$ (the non-strict version $\leq$ is defined as usual). We assume that $u \notin \mathcal{C} \cup \mathcal{F}$ and $+ \in \mathcal{F}$. First-order formulas are built in the usual way, defining $\top$, $\neg$, $\leftrightarrow$ as above.

² In HT_c , Boolean variables are already captured by truth values $\mathbf{t}$ and u (rather than $\mathbf{f}$ [alse]) (Cabalar et al. 2016).

³ We demonstrate that δ is a bijective function in the extended version of this paper (see <https://arxiv.org/abs/2507.13958>).

⁴ "Quantified" is used synonymously with "First-order".

The difference of our translation to Kamp's original one lies in the fact that our predicate names are not monadic and that we allow (partial) function symbols for simulating the valuation of a variable along time.

Given a temporal formula φ in THT_c , we define the first-order formula $\tau_t(\varphi)$ with the only free variable t as follows. For constraint atoms, our translation is defined as

$$\tau_t(c(\circ^{o_1}x_1, \dots, \circ^{o_n}x_n)) = p_c(f_{x_1}(t + o_1), \dots, f_{x_n}(t + o_n)).$$

where p_c is a predicate that simulates the behavior of c and each $f_{x_i} \in \mathcal{F}$ is a partial evaluable function associated with the variable x_i for $1 \leq i \leq n$. The value of a variable x at a time point t is given by $f_x(t)$. To capture the temporal offset i in $\circ^i x$, we shift the argument of f_x , resulting in $f_x(t + i)$. Note that f_x is partial because its argument might extend beyond the defined time points in a trace.

The rest of the translation follows the one by Kamp (1968):

$$\begin{aligned} \tau_t(\perp) &= \perp \\ \tau_t(\varphi \wedge \psi) &= \tau_t(\varphi) \wedge \tau_t(\psi) \\ \tau_t(\varphi \vee \psi) &= \tau_t(\varphi) \vee \tau_t(\psi) \\ \tau_t(\varphi \rightarrow \psi) &= \tau_t(\varphi) \rightarrow \tau_t(\psi) \\ \tau_t(\circ \varphi) &= \exists t'(t' = t + 1) \wedge \tau_{t'}(\varphi) \\ \tau_t(\varphi \mathbb{U} \psi) &= \exists t'(t \leq t' \wedge \tau_{t'}(\psi) \wedge (\forall t''(t \leq t'' \wedge t'' < t' \rightarrow \tau_{t''}(\varphi))) \\ \tau_t(\varphi \mathbb{R} \psi) &= \forall t'(t \leq t' \rightarrow (\tau_{t'}(\psi) \vee (\exists t''(t \leq t'' \wedge t'' < t' \wedge \tau_{t''}(\varphi)))) \\ \tau_t(\bullet \varphi) &= \exists t'(t = t' + 1) \wedge \tau_{t'}(\varphi) \\ \tau_t(\varphi \mathbf{S} \psi) &= \exists t'(t' \leq t \wedge \tau_{t'}(\psi) \wedge (\forall t''(t' < t'' \wedge t'' \leq t \rightarrow \tau_{t''}(\varphi))) \\ \tau_t(\varphi \mathbf{T} \psi) &= \forall t'(t' \leq t \rightarrow (\tau_{t'}(\psi) \vee (\exists t''(t' < t'' \wedge t'' \leq t \wedge \tau_{t''}(\varphi)))) \end{aligned}$$

The translation of derived operators can be done unfolding their definitions or using the shorter, equivalent formulas:

$$\begin{aligned} \tau_t(\mathbf{I}) &= \neg \exists t'(t' < t) & \tau_t(\mathbb{F}) &= \neg \exists t'(t < t') \\ \tau_t(\hat{\bullet} \psi) &= \forall t'(t = t' + 1 \rightarrow \tau_{t'}(\psi)) & \tau_t(\hat{\circ} \psi) &= \forall t'(t' = t + 1 \rightarrow \tau_{t'}(\psi)) \\ \tau_t(\diamond \psi) &= \exists t'(t \leq t' \wedge \tau_{t'}(\psi)) & \tau_t(\blacklozenge \psi) &= \exists t'(t' \leq t \wedge \tau_{t'}(\psi)) \\ \tau_t(\Box \psi) &= \forall t'(t \leq t' \rightarrow \tau_{t'}(\psi)) & \tau_t(\blacksquare \psi) &= \forall t'(t' \leq t \rightarrow \tau_{t'}(\psi)) \end{aligned}$$

As an example, let us consider the formula $\Box \blacklozenge (\circ^2 x = x)$ and the free variable t . The result $\tau_t(\Box \blacklozenge (\circ^2 x = x))$ of translating this formula is as follows:

$$\begin{aligned} \tau_t(\Box \blacklozenge (\circ^2 x = x)) &= \forall t'(t \leq t' \rightarrow \tau_{t'}(\blacklozenge (\circ^2 x = x))) \\ &= \forall t'(t \leq t' \rightarrow (\exists t''(t'' \leq t' \wedge \tau_{t''}(\circ^2 x = x)))) \\ &= \forall t'(t \leq t' \rightarrow (\exists t''(t'' \leq t' \wedge (f_x(t'' + 2) = f_x(t''))))) \end{aligned}$$

In our non-classical framework, the interpretation of first-order formulas requires Quantified HT Logic with evaluable functions (Cabalar 2011). In preparation for establishing the correctness of our translation in Proposition 6, we proceed to introduce the semantics of this logic. Given signature $\langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$, we define:

- $\mathcal{T}(\mathcal{C})$ as the set of all ground terms over $\mathcal{C}$,
- $\mathcal{T}(\mathcal{C} \cup \mathcal{F})$ as the set of all ground terms over $\mathcal{C}$ and $\mathcal{F}$,
- $\mathcal{A}(\mathcal{C}, \mathcal{P})$ as the set of all ground atoms over $\mathcal{C}$ and $\mathcal{P}$.

A *state* over $\langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ is a pair (σ, A) , where $A \subseteq \mathcal{A}(\mathcal{C}, \mathcal{P})$ and $\sigma : \mathcal{T}(\mathcal{C} \cup \mathcal{F}) \rightarrow \mathcal{T}(\mathcal{C}) \cup \{u\}$ is a function such that

1. $\sigma(s) = s$ for all $s \in \mathcal{T}(\mathcal{C})$
2. $\sigma(f(s_1, \dots, s_n)) = \begin{cases} u & \text{if } \sigma(s_i) = u \text{ for some } i \in \{1, \dots, n\} \\ \sigma(f(\sigma(s_1), \dots, \sigma(s_n))) & \text{otherwise} \end{cases}$

Given two states $S = (\sigma, A)$ and $S' = (\sigma', A')$, we write $S \preceq S'$ when both

1. $A \subseteq A'$ and
2. $\sigma(s) = \sigma'(s)$ or $\sigma(s) = u$ for all $s \in \mathcal{T}(\mathcal{C} \cup \mathcal{F})$.

We also write $S \prec S'$ when $S \preceq S'$ but $S \neq S'$.

Following Cabalar (2011), a $\text{QHT}_{\mathcal{F}}^{\preceq}(<)$ interpretation is a structure $\langle S_h, S_t \rangle$, where $S_h = (\sigma_h, I_h)$ and $S_t = (\sigma_t, I_t)$ such that $S_h \preceq S_t$. Given a first-order formula φ and a $\text{QHT}_{\mathcal{F}}^{\preceq}(<)$ interpretation $\mathcal{M} = \langle S_h, S_t \rangle$, we define the satisfaction relation as follows:

1. $\mathcal{M} \models \top$ and $\mathcal{M} \not\models \perp$
2. $\mathcal{M} \models p(s_1, \dots, s_n)$ if $p(\sigma_h(s_1), \dots, \sigma_h(s_n)) \in I_h$
3. $\mathcal{M} \models s = s'$ if $\sigma_h(s) = \sigma_h(s') \neq u$
4. $\mathcal{M} \models s < s'$ if $u \neq \sigma_h(s) < \sigma_h(s') \neq u$
5. $\mathcal{M} \models \varphi \wedge \psi$ if $\mathcal{M} \models \varphi$ and $\mathcal{M} \models \psi$
6. $\mathcal{M} \models \varphi \vee \psi$ if $\mathcal{M} \models \varphi$ or $\mathcal{M} \models \psi$
7. $\mathcal{M} \models \varphi \rightarrow \psi$ if $\mathcal{M}' \not\models \varphi$ or $\mathcal{M}' \models \psi$ for all $\mathcal{M}' \in \{\mathcal{M}, (S_t, S_h)\}$
8. $\mathcal{M} \models \exists x \varphi(x)$ if $\mathcal{M} \models \varphi(c)$ for some $c \in \mathcal{T}(\mathcal{C})$
9. $\mathcal{M} \models \forall x \varphi(x)$ if $\mathcal{M} \models \varphi(c)$ for all $c \in \mathcal{T}(\mathcal{C})$.

A $\text{QHT}_{\mathcal{F}}^{\preceq}(<)$ interpretation $\mathcal{M}$ is a $\text{QHT}_{\mathcal{F}}^{\preceq}(<)$ model of a formula φ when $\mathcal{M} \models \varphi$. A $\text{QHT}_{\mathcal{F}}^{\preceq}(<)$ model $\langle S_t, S_h \rangle$ is an equilibrium model for a formula φ , if there is no state S_h such that $S_h \prec S_t$ and $\langle S_h, S_t \rangle$ is a $\text{QHT}_{\mathcal{F}}^{\preceq}(<)$ model of φ .

We are now ready to define the model correspondence between HT_c traces and $\text{QHT}_{\mathcal{F}}^{\preceq}(<)$ interpretations. An HT_c trace $\mathbf{M} = \langle \mathbf{v}_h, \mathbf{v}_t \rangle$ of length λ for signature $\langle \mathcal{X}, \mathcal{D}, \mathcal{A} \rangle$ corresponds to a $\text{QHT}_{\mathcal{F}}^{\preceq}(<)$ interpretation $\mathcal{M} = \langle (\sigma_h, I_h), (\sigma_t, I_t) \rangle$ if for each $i \in [0..\lambda)$ and for each variable $x \in \mathcal{X}$, we get $\sigma_h(f_x(i)) = v_{h,i}(x)$, $\sigma_t(f_x(i)) = v_{t,i}(x)$,

$$\begin{aligned} I_h &= \{p_c(a_1, \dots, a_n) \mid c(\circ^{o_1}x_1, \dots, \circ^{o_n}x_n) \in \mathcal{A}, i \in [0..\lambda) \text{ and} \\ &\quad c(v_{h,i+o_1}(x_1), \dots, v_{h,i+o_n}(x_n)) = c(a_1, \dots, a_n) \in [c(\circ^{o_1}x_1, \dots, \circ^{o_n}x_n)]\} \text{ and} \\ I_t &= \{p_c(a_1, \dots, a_n) \mid c(\circ^{o_1}x_1, \dots, \circ^{o_n}x_n) \in \mathcal{A}, i \in [0..\lambda) \text{ and} \\ &\quad c(v_{t,i+o_1}(x_1), \dots, v_{t,i+o_n}(x_n)) = c(a_1, \dots, a_n) \in [c(\circ^{o_1}x_1, \dots, \circ^{o_n}x_n)]\}, \end{aligned}$$

where $[c(\circ^{o_1}x_1, \dots, \circ^{o_n}x_n)]$ is strict.

This model correspondence allows us to translate THT_c into $\text{QHT}_{\mathcal{F}}^{\preceq}(<)$. We let $\varphi[t/i]$ stand for the result of replacing each occurrence of variable t in φ by i .

Proposition 6.

Let $\mathbf{M} = \langle \mathbf{v}_h, \mathbf{v}_t \rangle$ be a HT_c trace and let $\mathcal{M} = \langle (\sigma_h, I_h), (\sigma_t, I_t) \rangle$ be its corresponding $\text{QHT}_{\mathcal{F}}^{\equiv}(<)$ interpretation. For all $i \in [0.. \lambda)$ and every temporal formula φ , we have $\mathbf{M}, i \models \varphi$ if $\mathcal{M} \models \tau_t(\varphi)[t/i]$.

In other words, we can consider THT_c as a subclass of theories in the fragment of $\text{QHT}_{\mathcal{F}}^{\equiv}(<)$ with partial evaluable functions. We show next that this correspondence is still valid when considering equilibrium models.

Proposition 7.

Let $\mathbf{M} = \langle \mathbf{v}_h, \mathbf{v}_t \rangle$ be a HT_c trace and let $\mathcal{M} = \langle (\sigma_h, I_h), (\sigma_t, I_t) \rangle$ be its corresponding $\text{QHT}_{\mathcal{F}}^{\equiv}(<)$ interpretation. For every temporal formula φ , $\mathbf{M}$ is an equilibrium model of φ if $\mathcal{M}$ is an equilibrium model of $\tau_t(\varphi)[t/0]$.

An important consequence of our translation is that it opens the possibility of applying first-order theorem provers for solving inference problems for THT_c . For instance, one salient THT_c inference problem is deciding THT_c -equivalence of two temporal theories with constraints, Γ_1 and Γ_2 , since it is a sufficient condition for their *strong equivalence* (Lifschitz et al. 2007), something that guarantees a safe replacement of Γ_1 by Γ_2 (or vice versa) in any arbitrary context. The translation of the $\text{QHT}_{\mathcal{F}}^{\equiv}(<)$ -formula obtained from $\tau_t(\cdot)$ into classical First-Order Logic can be done in two steps: first removing the partial functions in favor of predicates (Cabalar 2011), and second passing from quantified HT (without partial functions) into first-order classical logic (Pearce 2006).

4 Logic programs with temporal linear constraints

In this section, we consider the fragment of THT_c interpreted over linear constraints whose solution table is strict. We define a *linear term* as an expression of the form

$$d_1 \cdot \circ^{o_1} x_1 + \dots + d_n \cdot \circ^{o_n} x_n \quad (1)$$

where $d_i \in \mathbb{Z}$, $x_i \in \mathcal{X}$, and $o_i \in \mathbb{Z}$ for $0 \leq i \leq n$. Multiplication and addition are denoted by “ $\cdot$ ” and “ $+$ ”, respectively. An implicit multiplicative factor of 1 is assumed for standalone numbers d_i . Negative constants are represented using “ $-$ ”, and the “ $\cdot$ ” symbol may be omitted when contextually clear. Given a linear term α as in (1), we define $\text{terms}(\alpha) = \{\circ^{o_i} x_i \mid 0 \leq i \leq n\}$. For instance, $\text{terms}(2 \cdot \circ^{-2} x) = \{\circ^{-2} x\} = \{\bullet\bullet x\}$.

A *linear temporal constraint atom* is a temporal constraint atom of the form $\alpha \leq \beta$, where α and β are linear terms. We use the abbreviation $\alpha = \beta$, $\alpha < \beta$, and $\alpha \neq \beta$ for $(\alpha \leq \beta) \wedge (\beta \leq \alpha)$, $(\alpha \leq \beta) \wedge \neg(\beta \leq \alpha)$, and $(\alpha < \beta) \vee (\beta < \alpha)$, respectively. Similarly, $\text{terms}(\alpha \otimes \beta) = \text{terms}(\alpha) \cup \text{terms}(\beta)$, for $\otimes \in \{\leq, <, =, \neq\}$. Given our focus on strict solution tables, we have $[\alpha \leq \beta] \subseteq \mathcal{D}^n$ for all $\alpha \leq \beta$, where n is the number of terms in α and β . More generally, given a constraint atom c with a strict relation, we define its *complement* constraint as $[\bar{c}] = \mathcal{D}^n \setminus [c]$ (which is also strict). The interaction of complementary constraints with logical negation is given in the next proposition.

Proposition 8.

For any HT_c trace $\mathbf{M} = \langle \mathbf{v}_h, \mathbf{v}_t \rangle$ of length λ and any pair $c, \bar{c}$ of complementary constraints, with both $[c]$ and $[\bar{c}]$ being strict, we have that $\mathbf{M}, i \models c$ implies $\mathbf{M}, i \models \neg \bar{c}$ for all $i \in [0.. \lambda)$.

We thus obtain that $\alpha = \beta$ implies $\neg(\alpha \neq \beta)$ but not necessarily vice versa.

For illustration, consider the atoms $\circ x = x$ and $\circ x \neq x$, where x takes its values in $\mathbb{N}$. We define the solution relation for each constraint as

$$\begin{aligned} [\circ x = x] &= \{(a_1, a_2) \in \mathbb{N}^2 \mid a_1 = a_2 \neq u\} \\ [\circ x \neq x] &= \{(a_1, a_2) \in \mathbb{N}^2 \mid a_1 \neq a_2 \text{ and } a_1 \neq u \text{ and } a_2 \neq u\}. \end{aligned}$$

We have that $[\circ x \neq x] = \mathbb{N}^2 \setminus [\circ x = x]$, and therefore $[\circ x \neq x] = [\neg \circ x = x]$. By Proposition 8, $\circ x = x$ implies $\neg(\circ x \neq x)$. Conversely, let $\mathbf{M} = \langle \mathbf{v}_h, \mathbf{v}_t \rangle$ be an HT_c trace of length 2 defined as $v_{w,i}(x) = u$ for $w \in \{h, t\}$ and $i \in \{0, 1\}$. Then, we have that $\mathbf{M}, 0 \models \neg(\circ x \neq x)$ but $\mathbf{M}, 0 \not\models \circ x = x$.

When HT_c is extended with linear constraints, Cabalar *et al.* (2016) introduce an interesting feature known as *assignments*. This assignment operator is a specialized form of equality that allows us to check if a variable has a defined value. To illustrate this, consider the two following formulas:⁵

$$(x = y) \text{ and } (p = \mathbf{t} \rightarrow y = 10) \quad (2)$$

The equality $x = y$ implies that any assignment to x must also be an assignment to y , and vice versa. Given that $p = \mathbf{t}$ is not derivable, y appears to be undefined in this context. However, due to the strict nature of $[x = y]$, this allows for both x and y to take on any value, even if y seems undefined. To ensure that x is only assigned a value when y is defined, we can replace $x = y$ in (2) with:

$$y \leq y \rightarrow x = y \quad (3)$$

Because $y \leq y$ cannot be established (due to the underivability of $p = \mathbf{t}$ making y undefined), the resulting HT_c model has x , y and p all undefined. Indeed, within HT_c , the modified formula in (3) effectively represents the assignment $x := y$.

The remainder of this section focuses on extending this assignment mechanism of HT_c to our temporal setting in THT_c . For any linear expression α , we define

$$df(\alpha) \stackrel{\text{def}}{=} \bigwedge_{\circ^l x \in \text{terms}(\alpha)} \circ^l x \leq \circ^l x.$$

The satisfiability of $df(\alpha)$ necessitates that all its variables are defined, as shown next.

Proposition 9.

For all HT_c traces $\mathbf{M} = \langle (v_{h,i})_{i \in [0..\lambda)}, (v_{t,i})_{i \in [0..\lambda)} \rangle$ and for all linear expressions α , the following statements are equivalent for all $i \in [0..\lambda)$

1. $\mathbf{M}, i \models df(\alpha)$
2. $v_{h,i}(\circ^l x) = v_{t,i}(\circ^l x) \neq u$ for all $\circ^l x \in \text{terms}(\alpha)$

In our context, an *assignment* for a variable $x \in \mathcal{X}$ is an expression of the form $\circ^l x := \alpha$, where α is a linear temporal term. The assignment operator can be seen as a derived

⁵ Recall that atom $p = \mathbf{t}$ is the constraint representation of variable p (see Section 2.2).

operator defined as

$$\circ^l x := \alpha \stackrel{\text{def}}{=} (df(\alpha) \rightarrow \circ^l x = \alpha).$$

With it, we now define a syntactic subclass that takes the form of a logic program. To ease notation, we let $(\varphi \leftarrow \psi) \stackrel{\text{def}}{=} (\psi \rightarrow \varphi)$ and $(\varphi, \psi) \stackrel{\text{def}}{=} (\psi \wedge \varphi)$. Then, a *temporal linear constraint rule* (or just *rule*) is of the form:

$$\circ^l x := \alpha \leftarrow \ell_1, \dots, \ell_m, \neg \ell_{m+1}, \dots, \neg \ell_k \quad (4)$$

where $\circ^l x := \alpha$ is an assignment and each ℓ_i is a linear temporal constraint atom for $0 \leq m \leq k$. Placing Rule (4) under the $\square$ operator ensures its satisfaction across all temporal states. As with HT_c , assignments appearing in rule heads can be transformed into an analogous logic programming representation.

Proposition 10.

A rule of form (4) is equivalent to the rule

$$\circ^l x = \alpha \leftarrow \circ^{\alpha_1} x_1 \leq \circ^{\alpha_1} x_1, \dots, \circ^{\alpha_n} x_n \leq \circ^{\alpha_n} x_n, \ell_1, \dots, \ell_m, \neg \ell_{m+1}, \dots, \neg \ell_k.$$

where $\circ^{\alpha_i} x_i \in \text{terms}(\alpha)$ for $0 \leq i \leq n = |\text{terms}(\alpha)|$.

The logicprogramming fragment of THT_c offers a powerful approach to modeling various aspects of dynamic systems, including the representation of inertia rules and default values. To demonstrate the expressive power of our formalism, let us reconsider our initial scenario: “A radar is positioned at the 400 km mark on a road with a speed limit of 90 km/h. A car is initially traveling at 80 km/h. At time instant 4, the car accelerates by 11.35 km/h. Subsequently, at time instant 6, it decelerates by 2.301 km/h. The problem is to determine whether the car will exceed the speed limit and thus incur a fine.”

To formalize this scenario within THT_c , we introduce the following variables. The numerical fluents p and s (ranging over $\mathbb{N}$) represent the car’s position and speed, respectively. The numerical variables $rdpos$ and $rdlimit$ (ranging over $\mathbb{N}$) refer to the radar’s position and speed limit. The numerical action acc (ranging over $\mathbb{Z}$) models the car’s acceleration (positive) or deceleration (negative). Lastly, the Boolean variable $fine$ indicates whether the car’s speed exceeded $rdlimit$ at $rdpos$, leading to a fine.

The subsequent set of rules formalizes our scenario in THT_c .⁶

$$p := 0 \quad (5)$$

$$s := 80 \quad (6)$$

$$\square(rdlimit := 90) \quad (7)$$

$$\square(rdpos := 400) \quad (8)$$

$$\square(\circ s := s + acc) \quad (9)$$

$$\square(\circ s := s \leftarrow \neg(\circ s \neq s)) \quad (10)$$

$$\square(\circ p := p + s) \quad (11)$$

$$\square(\circ fine \leftarrow p < rdpos, \circ p \geq rdpos, \circ s > rdlimit) \quad (12)$$

⁶ To facilitate understanding, the values of s , acc , and pos are presented using decimal notation. In a real-world implementation, speed and acceleration should be expressed in m/h , while position should be expressed in meters.

Rules preceded by the always operator $\Box$ in our THT_c formalization are enforced throughout the entire temporal evolution, contrasting with Rules (5) and (6), which are specific to the initial state. Specifically, Rules (5) and (6) set the car's initial position to 0 meters and its speed to 80 km/h. The radar's characteristics, its position at 400 km and a constant speed limit of 90 km/h, are defined by Rules (7) and (8). The dynamics of the car's speed are captured by Rules (9) and (10): the speed remains unchanged unless an acceleration value (acc) is present, causing a corresponding adjustment. Rule (11) dictates that the car's position changes based on its speed. The condition for receiving a fine is formalized in Rule (12): if the car's speed (s) exceeds the radar's speed limit ($rdlimit$) at or after passing the radar's position ($rdpos$) at any time, a fine is issued. Finally, to model the acceleration of 11.350 km/h at time 4 and the deceleration of 2.301 km/h at time 6, we include the following temporal assignments:

$$\circ^4 acc := 11.35 \qquad \qquad \qquad \circ^6 acc := -2.301$$

These two rules for acceleration and deceleration are applied specifically at time points 4 and 6, respectively. Note that our formalization does not provide a value for acc in the initial state. Since no value is provided, acc is left undefined at the initial state, Rule (9) is inapplicable and the inertia rule in (10) keeps the speed (s) constant. However, at time points 4 and 6, a value for acc is given, so Rule (9) applies and the value of s is modified.

An alternative formalization could rely on the use of the default rule

$$\Box (acc := 0 \leftarrow \neg (acc \neq 0))$$

to set the value of acc to 0 in the absence of information. Under the addition of such a rule, inertia (10) could be removed, since it becomes redundant: when there is no acceleration, $acc = 0$ and we already derive $\circ s = s$ through (9).

The following table presents an equilibrium model that satisfies the representation of our scenario. The values for $rdlimit$ and $rdpos$ are omitted from the table as they remain constant over time. The evolution of the remaining fluents is as follows:

<i>time</i>	<i>s(km/h)</i>	<i>p(km)</i>	<i>acc(km/h)</i>
0	80	0	<i>u</i>
1	80	80	<i>u</i>
2	80	160	<i>u</i>
3	80	240	<i>u</i>
4	80	320	11.35
5	91.35	400	<i>u</i>
6	91.35	491.35	-2.301
7	89.049	582.7	<i>u</i>
8	89.049	671.749	<i>u</i>

Between time points 4 and 5, the car's speed exceeds the limit as it passes the radar's position, resulting in the *fine* atom becoming true at time point 5.

5 Discussion

We have integrated two established extensions of HT, namely THT and HT_c (along with their equilibrium variants), into a unified framework for temporal (nonmonotonic) reasoning with constraints. Our work is inspired by extensions of (monotonic) Linear Temporal Logic (LTL) interpreted over constraint systems and first-order theories, in particular the approaches taken by Demri (2006a) and Geatti *et al.* (2022), which apply next and previous operators to non-Boolean variables. In many of these cases, the constraints considered are based on Presburger arithmetic (Demri 2006b) or qualitative spatial formalisms (Vilain and Kautz 1986; Wolter and Zakharyashev 2000; Balbiani and Condotta 2001). Within the ASP paradigm, a significant early extension of temporal answer sets with constraints was introduced by Giordano *et al.* (2013a), building upon their earlier work (Giordano *et al.* 2013b). This approach utilizes logic programming syntax and semantics, which differs from our HT-based approach. Furthermore, their constraints typically refer to variables within individual states, unlike our approach which allows for the combination of variables across multiple temporal states.

In addition to defining and formally elaborating on the semantics of THT_c , we extended Kamp's translation (Kamp 1968) to our approach by mapping THT_c formulas into $\text{QHT}_{\overline{\mathcal{F}}}(<)$, and notably demonstrated that this correspondence extends to the respective equilibrium models. We further investigated the logic programming fragment of THT_c , illustrating its expressive power through modeling our initial scenario involving temporal numeric constraints. This work lays the groundwork for the future integration of constraint reasoning into other HT-based temporal extensions, such as dynamic and metric logics of HT (and their equilibrium variants) (Bosser *et al.* 2018; Becker *et al.* 2024b). Advancing this line of research will necessitate adapting existing computational methods (Becker *et al.* 2024a) to operate over constraints, extending beyond solely propositional atoms.

As a first step, we have only considered linear constraints. Future work includes studying the integration of periodicity constraints (Demri 2006b) and qualitative spatial constraints (Wolter and Zakharyashev 2000; Balbiani and Condotta 2001). Periodicity constraints would allow for expressing congruence relations among different variables, proving particularly useful for capturing cyclical behaviors or recurring events. Integrating spatial constraints would enable us to assign spatial meaning to variables and describe their dynamics, for instance, modeling objects that change their position or size over time.

In the classical setting, the computational complexity of temporal logics interpreted over constraint systems is often highly undecidable (Demri 2006a), with decidability strongly depending on the specific constraint system. For example, the satisfiability problem for LTL interpreted over constraint systems based on Presburger arithmetic (including periodicity constraints) is in PSPACE under certain completion properties (Balbiani and Condotta 2002; Demri 2006a). When considering qualitative spatial constraints expressed using the Region Connection Calculus, the problem is decidable, with complexity ranging from NP to EXPSpace, depending on the imposed restrictions (Wolter and Zakharyashev 2000). Following a strategy similar to (Cabalar and Demri 2011), it may be possible to establish a bijection between classical and equilibrium models,

potentially allowing us to establish a lower bound on the complexity of the satisfiability problem in THT_c . However, determining a corresponding upper bound remains an open challenge.

Acknowledgments

This work was supported by the DFG grant SCHA 550/15 (Germany), the Etoiles Montantes CTASP project (Region Pays de la Loire, France), and the MICIU/AEI/10.13039/501100011033 grant PID2023-148531NB-I00 (Spain).

Competing interests

The authors declare none.

References

- AGUADO, F., CABALAR, P., DIÉGUEZ, M., PÉREZ, G., SCHAUB, T., SCHUHMANN, A. and VIDAL, C. 2023. Linear-time temporal answer set programming. *Theory and Practice of Logic Programming* 23, 1, 2–56. <https://doi.org/10.1017/S1471068421000557>.
- BALBIANI, P. and CONDOTTA, J. 2001. Une logique pour le raisonnement spatio-temporel basée sur PLTL et l'algèbre des rectangles. In *Journées Nationales sur les Modèles de Raisonnement (JNMR'01)*, 9–22.
- BALBIANI, P. and CONDOTTA, J. 2002. Computational complexity of propositional linear temporal logics based on qualitative spatial or temporal reasoning. In *Proceedings of the Fourth International Workshop on Frontiers of Combining Systems (FroCoS'02)*, A. ARMANDO, Ed. Vol. 2309 of *Lecture Notes in Computer Science*, Springer-Verlag, 162–176.
- BECKER, A., CABALAR, P., DIÉGUEZ, M., HAHN, S., ROMERO, J. and SCHAUB, T. 2024a. Compiling metric temporal answer set programming. In *Proceedings of the Seventeenth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'24)*, C DODARO, G GUPTA and M. V. MARTINEZ, Eds. Vol. 15245 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 15–29. https://doi.org/10.1007/978-3-031-74209-5_2.
- BECKER, A., CABALAR, P., DIÉGUEZ, M., SCHAUB, T. and SCHUHMANN, A. 2024b. Metric temporal answer set programming over timed traces. *Theory and Practice of Logic Programming* 24, 3, 425–452. <https://doi.org/10.1017/S1471068424000139>.
- BOSSER, A., CABALAR, P., DIÉGUEZ, M. and SCHAUB, T. 2018. Introducing temporal stable models for linear dynamic logic. In *Proceedings of the Sixteenth International Conference on Principles of Knowledge Representation and Reasoning (KR'18)*, M. THIELSCHER, F. TONI and F. WOLTER, Eds. AAAI Press, 12–21.
- CABALAR, P. 2011. Functional answer set programming. *Theory and Practice of Logic Programming* 11, 2-3, 203–233. <https://doi.org/10.1017/S1471068410000517>.
- CABALAR, P. and DEMRI, S. 2011. Automata-based computation of temporal equilibrium models. In *Proceedings of the Twenty-first International Symposium on Logic-Based Program Synthesis and Transformation (LOPSTR'11)*, G. VIDAL, Ed. Vol. 7225 of *Lecture Notes in Computer Science*, Springer-Verlag, 57–72.
- CABALAR, P., KAMINSKI, R., OSTROWSKI, M. and SCHAUB, T. 2016. An ASP semantics for default reasoning with constraints. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16)*, S. KAMBHAMPATI, Ed. IJCAI/AAAI Press, 1015–1021. <https://doi.org/10.5555/3060621.3060762>

- DECHTER, R. 2003. *Constraint Processing*. Morgan Kaufmann Publishers.
- DEMRI, S. 2006a. Linear-time temporal logics with Presburger constraints: An overview. *Journal of Applied Non-Classical Logics* 16, 3-4, 311–348.
- DEMRI, S. 2006b. LTL over integer periodicity constraints. *Theoretical Computer Science* 360, 1-3, 96–123.
- GEATTI, L., GIANOLA, A. and GIGANTE, N. 2022. Linear temporal logic modulo theories over finite traces. In *Proceedings of the Thirty-first International Joint Conference on Artificial Intelligence (IJCAI'22)*, L. DE RAEDT, Ed. ijcai.org, 2641–2647.
- GIORDANO, L., MARTELLI, A. and DUPRÉ, D. T. 2013b. Reasoning about actions with temporal answer sets. *Theory and Practice of Logic Programming* 13, 2, 201–225.
- GIORDANO, L., MARTELLI, A., SPIOTTA, M. and DUPRÉ, D. T. 2013a. Business process verification with constraint temporal answer set programming. *Theory and Practice of Logic Programming* 13, 4-5, 641–655. <https://doi.org/10.1017/S1471068413000409>.
- KAMP, J. 1968. Tense Logic and the Theory of Linear Order. *PhD thesis*, University of California at Los Angeles.
- LIFSCHITZ, V. 2008. What is answer set programming? In *Proceedings of the Twenty-third National Conference on Artificial Intelligence (AAAI'08)*, D. FOX and C. GOMES, Eds. AAAI Press, 1594–1597.
- LIFSCHITZ, V., PEARCE, D. and VALVERDE, A. 2007. A characterization of strong equivalence for logic programs with variables. In *Proceedings of the Ninth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'07)*, C. BARAL, G. BREWKA and J. SCHLIPF, Eds. Vol. 4483 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 188–200.
- PEARCE, D. 2006. Equilibrium logic. *Annals of Mathematics and Artificial Intelligence* 47, 1-2, 3–41. <https://doi.org/10.1007/s10472-006-9028-z>.
- VILAIN, M. and KAUTZ, H. 1986. Constraint propagation algorithms for temporal reasoning. In *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI'86)*, T. KEHLER, Ed. Morgan Kaufmann Publishers, 377–382.
- WOLTER, F. and ZAKHARYASCHEV, M. 2000. Spatio-temporal representation and reasoning based on RCC-8. In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'00)*, A. COHN, F. GIUNCHIGLIA and B. SELMAN, Eds. Morgan Kaufmann Publishers, 3–14.