Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning
Main Track

On the Effects of Adding Assignments in Linear-Time Temporal Logics Modulo Theories

Stéphane Demri[1] , **Raul Fervari**[1,2,3]
[1]Université Paris-Saclay, CNRS, ENS Paris-Saclay, Laboratoire Méthodes Formelles, 91190, Gif-Sur-Yvette, France
[2]Universidad Nacional de Córdoba, FAMAF, Argentina
[3]Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina

Abstract

We introduce linear-time temporal logics with past operators featuring a simple assignment modality that performs local changes on the models. Such structures are infinite sequences of valuations interpreting variables by elements from a possibly infinite data domain. We study several fragments as well as the case with the Boolean domain, for which we establish that it is actually as expressive as first-order logic over infinite sequences of propositional valuations. For the logics over concrete domains $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{Q}$ equipped with the respective linear ordering and equality tests, we show the satisfiability problem is decidable, and that the logics are as expressive as the version without the assignment operator. Interestingly, this entails such assignments provide a huge conciseness, which is then helpful for succinct specifications.

1 Introduction

Temporal logics with concrete domains. Enriching models with data values is a natural process that has led to the study of logics over the so-called concrete domains. In such logics, constraints about the data values are part of the formulae. Prominent examples include description logics with concrete domains, see e.g. (Lutz 2002a; Lutz 2002b; Labai, Ortiz, and Simkus 2020; Baader and Rydval 2022), temporal logics with concrete domains, see e.g. (Balbiani and Condotta 2002; Carapelle 2015), and more recently the so-called LTL modulo theories, see e.g. (Felli, Montali, and Winkler 2022; Cimatti et al. 2022; Geatti et al. 2023; Geatti, Gianola, and Gigante 2025; Beutner and Finkbeiner 2025). Considering data values is a natural perspective to reason about concrete information (like a person's age or the distance between two locations), see the seminal paper (Baader and Hanschke 1991), and about data structures in general (a string, a record, etc). This framework hosts many decision problems such as satisfiability, model-checking, realizability and runtime monitoring, to cite a few examples, see e.g. (Bhaskar and Praveen 2024; Faella and Parlato 2024; Rodríguez and Sánchez 2024; Gianola, Montali, and Winkler 2025).

Changing models. While the generalisation of propositional variables to constraints interpreted on domains better reflects the ubiquity of data structures, other investigations deal with the ability to *modify* the models, such as in public announcement logics (Lutz 2006), in sabotage modal logics (Aucher, van Benthem, and Grossi 2018), in alternating-time temporal logics (Catta et al. 2024; Galimullin et al. 2025) or in modal logics with graph modifiers (Aucher et al. 2009; Areces, Fervari, and Hoffmann 2015). Other approaches focus on updating the value of a variable, as e.g. in formalisms with the freeze binding mechanism. Typically, a formula $\downarrow_{r=\mathtt{x}} \varphi$ states that freezing the current value of $\mathtt{x}$ in the register r, makes φ true. This mechanism can be traced back to works about real-time logics, see e.g. (Alur and Henzinger 1994), modal hybrid logics, see e.g. (Areces and ten Cate 2007), logics for data trees, see e.g. (Figueira 2010), and modal logics with λ-abstraction (Fitting 2002).

Our motivations. In this paper, we aim at understanding logical formalisms that combine theory reasoning with the ability to update the model *atomically* and *locally* by replacing a value at the current position by a local value (available at bounded distance). Typically, we would like to handle properties of the form "*after replacing the current value of* $\mathtt{x}$ *by the previous value of* $\mathtt{y}$*, the formula* φ *holds true.*" From the perspective of formal verification of programs, it is natural to consider methods in which explicit assignments to variables are allowed. With a similar goal in mind, different approaches on this problem have been proposed, such as the Dynamic Logics with Propositional Assignments, a family of logics designed to reason about different mental/cognitive attitudes, including planning scenarios (Herzig, Maris, and Vianey 2019), and belief revision models (Herzig 2014). Another example is that of (Belardinelli et al. 2023; Belardinelli et al. 2025), extending Epistemic Logic with program assignments, tailored to modelling the global state of a (partially observed) set of variables, unlike here in which changes are local. Herein, we would like the assignment operators to perform local changes in the models made of ω-sequences of valuations $\mathsf{Var} \to \mathbb{D}$, where $\mathbb{D}$ is a domain equipped with relations. Unlike most assignments in the literature with global effects, see e.g. (van Ditmarsch, Herzig, and de Lima 2012; Galimullin et al. 2025), *only one* value is changed at once. In our case, computational difficulties arise because the domain $\mathbb{D}$ is possibly infinite and the position $i \in \mathbb{N}$ where the replacement is performed is not stored. We would like to design linear-time temporal logics modulo theories in which the assignment operator leads to a decidable satisfiability problem. More generally,

we investigate decidability, complexity and expressivity issues for these logics. Consequently, while we aim at studying formalisms that cannot be captured by Boolean abstractions (for instance with the domain $\mathbb{N}$), we wish to express concisely non-trivial model changes that inherently concern *data values* and *not the structure* of the models as it is usually done, see e.g. (Aucher et al. 2009). By way of example, to express the property *"The formula φ does not hold but changing somewhere the value of* $\mathtt{x}$ *by 0 and the value of* $\mathtt{y}$ *by 1 leads to a model satisfying φ",* the formula

$$\neg\varphi \wedge \mathsf{F}\big(\langle \mathtt{x}{:=}0\rangle\,(\mathsf{E}\langle \mathtt{y}{:=}1\rangle \mathsf{P}(\neg \mathsf{X}^{\text{-}1}\top \wedge \varphi))\big)$$

could do the job assuming that $\mathsf{E}\psi$ stands for $\mathsf{P}\psi \vee \mathsf{F}\psi$, the connective F is the standard sometimes operator from LTL and P is its past counterpart. Above, $\neg \mathsf{X}^{\text{-}1}\top$ holds true only at the origin of the model. As far as we know, no logical formalism related to LTL modulo theories can express naturally this simple property. Still, this can be *encoded* within constraint PLTL (LTL with past) if the number of variables is doubled by considering the original model and the variant model after the two assignments. This works fine for this property but it may happen that the number of changes to be considered is unbounded, typically with a pattern of the form $\mathsf{G}\langle \mathtt{x}{:=}\mathtt{t}\rangle$. Moreover, doubling the number of variables corresponds to an implicit existential second-order quantification and we would like to express the property directly in the native language (see a similar situation in (Wolper 1983, Sec. 4) with the even property).

Our contributions. Concrete domains $\mathbb{D}$ are defined as sets equipped with relations. We introduce the logic $\mathsf{PLTL\text{-}A}(\mathbb{D})$ (parameterised by $\mathbb{D}$) that extends past-time linear-time temporal logic $\mathsf{PLTL}(\mathbb{D})$ (where atomic formulae are made of $\mathbb{D}$-constraints between local values), by adding unary modalities as in $\langle \mathtt{x}{:=}\mathsf{X}^n\mathtt{y}\rangle\varphi$. The latter is read as *"after the value of the variable* $\mathtt{x}$ *is replaced by the local value for* $\mathsf{X}^n\mathtt{y}$ *(past or future values of* $\mathtt{y}$ *depending whether* $n < 0$*), the formula φ holds"*. The variables are interpreted on the domain $\mathbb{D}$. We focus here on domains of the form $(\mathbb{D}, =, <, (=_d)_{d\in\mathbb{D}})$, where $=, <$ are the corresponding equality and 'less than' relations, and $=_d$ is an equality test with respect to the constant d. So typically, $\mathbb{D} \in \{\{0,1\}, \mathbb{N}, \mathbb{Z}, \mathbb{Q}\}$. Modalities of the form $\langle \mathtt{x}{:=}d\rangle$ for some $d \in \mathbb{D}$ are also allowed. To the best of our knowledge, data updates in logics with concrete domains have been seldom investigated, see an interesting example in (Bansal, Brochenin, and Lozes 2009) involving separation logic and dynamic data structures.

After providing reductions from the satisfiability problem for $\mathsf{PLTL\text{-}A}(\mathbb{D})$ to the problem for interesting strict fragments of $\mathsf{PLTL\text{-}A}(\mathbb{D})$ (typically by bounding the number of variables, or by restricting the shape of $\mathtt{t}$ in $\langle \mathtt{x}{:=}\mathtt{t}\rangle$), we start our investigations on the simple extension $\mathsf{PLTL\text{-}A}(\{0,1\})$ of PLTL (also written $\mathsf{PLTL}^{:=}$). We show that $\mathsf{PLTL}^{:=}$ is as expressive as PLTL by providing a translation into $\mathsf{FO}(\omega)$ and then evoking Kamp's Theorem (Kamp 1968; Diekert and Gastin 2008; Rabinovich 2014). We show that $\mathsf{PLTL}^{:=}$ is significantly more succinct than PLTL as we prove that for every $\mathbb{D} \in \{\{0,1\}, \mathbb{N}, \mathbb{Z}, \mathbb{Q}\}$, the satisfiability problem for $\mathsf{PLTL\text{-}A}(\mathbb{D})$ is Tower-hard (see (Schmitz 2016) for the definition of the complexity class Tower). Indeed, the problem is PSpace-complete for $\mathsf{PLTL}(\mathbb{D})$. Moreover, we show that for every $\mathbb{D} \in \{\{0,1\}, \mathbb{N}, \mathbb{Z}, \mathbb{Q}\}$, the satisfiability problem for $\mathsf{PLTL\text{-}A}(\mathbb{D})$ is decidable. To do so, we introduce a new first-order logic on multi-attributed data words (that we actually believe interesting for its own sake, see landmark logical formalisms in (Bojańczyk et al. 2011; Jurdziński and Lazić 2011; Decker et al. 2014)) that can express 'local' domain constraints from $\mathbb{D}$ (see Sec. 4.1). We establish that the satisfiability problem for this logic is decidable. The semantics for $\mathsf{PLTL\text{-}A}(\mathbb{D})$ can be internalised in this new first-order logic, thus obtaining decidability. It is worth noting that handling the concrete domains $\mathbb{N}$ and $\mathbb{Z}$ is already notoriously difficult *without* assignment operators, see e.g. (Segoufin and Toruńczyk 2011; Labai, Ortiz, and Simkus 2020). Our results allow us to conclude that the new assignment operators add conciseness but not expressivity. Finally, our results easily apply to variants of $\mathsf{PLTL\text{-}A}(\mathbb{D})$, e.g. on finite traces, and for other concrete domains including RCC8 and Allen's interval algebra, just to name a few examples.

2 PLTL With Constraints and Assignments

We start by introducing linear-time temporal logics extending LTL (Pnueli 1977; Gabbay et al. 1980) where propositional variables are replaced by symbolic constraints. Herein, the models are ω-sequences of valuations interpreting variables by elements in the domain $\mathbb{D}$, see e.g. (Balbiani and Condotta 2002). Furthermore, modalities $\langle \mathtt{x}{:=}\mathtt{t}\rangle$ are used to build formulae whose semantical counterpart is to perform a local change in the models.

2.1 The logics $\mathsf{PLTL\text{-}A}(\mathbb{D})$

Concrete domains. In what follows, a concrete domain is a tuple $(\mathbb{D}, =, <, (=_d)_{d\in\mathbb{D}})$, where $\mathbb{D}$ is a non-empty set of elements, $=$ is the equality relation on $\mathbb{D}$, and $<$ is a total strict ordering on $\mathbb{D}$. Moreover, $=_d$ is interpreted as the singleton set $\{d\}$, which is helpful to perform equality tests with respect to the constant d. We often simply write $\mathbb{D}$ instead of $(\mathbb{D}, =, <, (=_d)_{d\in\mathbb{D}})$.

Syntax. Let $\mathsf{Var} = \{\mathtt{x}, \mathtt{y}, \mathtt{z}, \ldots\}$ be a countably infinite set of variables. The set of formulae for the logic $\mathsf{PLTL\text{-}A}(\mathbb{D})$ is defined by the following grammar:

$$\varphi, \psi ::= \mathsf{X}^n\mathtt{x} \sim \mathsf{X}^m\mathtt{y} \mid \mathsf{X}^n\mathtt{x} = d \mid \neg\varphi \mid \varphi \vee \psi \mid \mathsf{X}\varphi \mid \mathsf{X}^{\text{-}1}\varphi \mid \varphi\mathsf{U}\psi \mid \varphi\mathsf{S}\psi \mid \langle \mathtt{x}{:=}\mathsf{X}^m\mathtt{y}\rangle\varphi \mid \langle \mathtt{x}{:=}d\rangle\varphi,$$

where $\mathtt{x}, \mathtt{y} \in \mathsf{Var}$, $n, m \in \mathbb{Z}$, $\sim \in \{=, <\}$, and $d \in \mathbb{D}$ (elements of the concrete domains $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\{0,1\}$ are encoded in binary, and similarly for any other arbitrary domain $\mathbb{D}$ considered in this paper). The expression X^n (resp. $\mathsf{X}^{\text{-}n}$) with $n \geqslant 0$ stands for the sequence of length n made of the symbol X (resp. $\mathsf{X}^{\text{-}1}$) only. This corresponds to a unary encoding. Expressions of the form $\mathsf{X}^n\mathtt{x}$ are sometimes called terms. Sometimes, we simply write $\mathsf{X}\mathtt{x}$ if $n = 1$, or just $\mathtt{x}$ if $n = 0$, both in terms and assignments. Other operators are defined as usual, in particular $\mathsf{X}^n\mathtt{x} \neq \mathtt{t} \stackrel{\text{def}}{=} \neg(\mathsf{X}^n\mathtt{x} = \mathtt{t})$ (with $\mathtt{t} \in \{\mathsf{X}^m\mathtt{y}, d\}$), $\mathsf{F}\varphi \stackrel{\text{def}}{=} \top\mathsf{U}\varphi$ and $\mathsf{G}\varphi \stackrel{\text{def}}{=} \neg\mathsf{F}\neg\varphi$. Finally, for all $n > 0$, we write $\mathsf{X}^n\varphi$ (resp. $\mathsf{X}^{\text{-}n}\varphi$) to denote a sequence made of n temporal operators X (resp. $\mathsf{X}^{\text{-}1}$).

Semantics. A **model for PLTL-A**$(\mathbb{D})$ is a map $\sigma : \mathbb{N} \to (\mathsf{Var} \to \mathbb{D})$. We write $\sigma(i)(\mathtt{x})$ to denote the value assigned to the variable $\mathtt{x}$ at the position i. Let σ be a model, and let $i \geqslant 0$, the **satisfaction relation** $\Vdash$ between σ, i and formulae is inductively defined as follows (below $\sim \in \{=, <\}$):

$$
\begin{array}{lcl}
\sigma, i \Vdash \mathsf{X}^n\mathtt{x} \sim \mathsf{X}^m\mathtt{y} & \stackrel{\text{def}}{\Leftrightarrow} & i + n \geqslant 0,\ i + m \geqslant 0 \text{ and } \sigma(i+n)(\mathtt{x}) \sim \sigma(i+m)(\mathtt{y}) \\
\sigma, i \Vdash \mathsf{X}^n\mathtt{x} = d & \stackrel{\text{def}}{\Leftrightarrow} & i + n \geqslant 0 \text{ and } \sigma(i+n)(\mathtt{x}) = d \\
\sigma, i \Vdash \neg\varphi & \stackrel{\text{def}}{\Leftrightarrow} & \sigma, i \nVdash \varphi \\
\sigma, i \Vdash \varphi \vee \psi & \stackrel{\text{def}}{\Leftrightarrow} & \sigma, i \Vdash \varphi \text{ or } \sigma, i \Vdash \psi \\
\sigma, i \Vdash \mathsf{X}\varphi & \stackrel{\text{def}}{\Leftrightarrow} & \sigma, i+1 \Vdash \varphi \\
\sigma, i \Vdash \mathsf{X}^{-1}\varphi & \stackrel{\text{def}}{\Leftrightarrow} & i > 0 \text{ and } \sigma, i-1 \Vdash \varphi \\
\sigma, i \Vdash \varphi \mathsf{U} \psi & \stackrel{\text{def}}{\Leftrightarrow} & \sigma, j \Vdash \psi \text{ for some } j \geqslant i \text{ s.t. } \sigma, k \Vdash \varphi \text{ for all } i \leqslant k < j \\
\sigma, i \Vdash \varphi \mathsf{S} \psi & \stackrel{\text{def}}{\Leftrightarrow} & \sigma, j \Vdash \psi \text{ for some } 0 \leqslant j \leqslant i \text{ s.t. } \sigma, k \Vdash \varphi \text{ for all } j < k \leqslant i \\
\sigma, i \Vdash \langle \mathtt{x}{:=}\mathsf{X}^n\mathtt{y} \rangle \varphi & \stackrel{\text{def}}{\Leftrightarrow} & i + n \geqslant 0 \text{ and } \sigma[\mathtt{x} \mapsto_i \sigma(i+n)(\mathtt{y})], i \Vdash \varphi \\
\sigma, i \Vdash \langle \mathtt{x}{:=}d \rangle \varphi & \stackrel{\text{def}}{\Leftrightarrow} & \sigma[\mathtt{x} \mapsto_i d], i \Vdash \varphi,
\end{array}
$$

where $\sigma[\mathtt{x} \mapsto_i d](j)(\mathtt{y}) = \sigma(j)(\mathtt{y})$ if $j \neq i$ or $\mathtt{x}$ is syntactically different from $\mathtt{y}$, and $\sigma[\mathtt{x} \mapsto_i d](\mathtt{x})(i) = d$. Note that data values at distinct positions can therefore be compared. A formula φ is **satisfiable** iff there is a model σ such that $\sigma, 0 \Vdash \varphi$. The interesting formulae $\langle \mathtt{x}{:=}\mathsf{X}^n\mathtt{y} \rangle \varphi$ are those containing past-time operators in φ as the current position may be revisited after the change of the value $\mathtt{x}$.

Example 1. *We provide here some examples of formulae.*

1. *The formula* $(\mathtt{x} = 0 \Rightarrow \mathsf{G}\neg\varphi) \wedge \langle \mathtt{x}{:=}\mathsf{X}\mathtt{x} \rangle \mathsf{F}\varphi$ *states that the current value of* $\mathtt{x}$ *equals* 0 *implies in the future* φ *never holds, but after updating the value of* $\mathtt{x}$ *by the value of* $\mathtt{x}$ *at the next position,* φ *holds eventually in the future.*
2. *The formula* $\mathsf{XG}(\langle \mathtt{x}{:=}\mathsf{X}^{-1}\mathtt{x} \rangle \varphi)$ *states that always in the future, changing the value of* $\mathtt{x}$ *to its previous value makes* φ *true.*
3. *Atomic formulae of the form* $\mathsf{X}^n\mathtt{x} \sim \mathsf{X}^m\mathtt{y}$ *and* $\mathsf{X}^n\mathtt{x} = d$ *are handy to express certain properties but we could restrict the language while keeping the same expressive power. Indeed,* $\mathsf{X}^n\mathtt{x} \sim \mathsf{X}^m\mathtt{y}$ *(resp.* $\mathsf{X}^n\mathtt{x} = d$*) is logically equivalent to* $\mathsf{X}^n\top \wedge \mathsf{X}^m\top \wedge \mathsf{X}^n(\mathtt{x} \sim \mathsf{X}^{m-n}\mathtt{y})$ *with* $n < m$ *(resp.* $\mathsf{X}^n(\mathtt{x} = d)$*).*

The satisfiability problem of PLTL-A$(\mathbb{D})$ without assignments, written PLTL$(\mathbb{D})$, has been well-studied.

Proposition 1. *(Sistla and Clarke 1985; Balbiani and Condotta 2002; Demri and Gascon 2008; Segoufin and Toruńczyk 2011) For all the domains* $\mathbb{D} \in \{\{0,1\}, \mathbb{N}, \mathbb{Z}, \mathbb{Q}\}$*, the satisfiability problem for* PLTL$(\mathbb{D})$ *is* PSpace*-complete.*

2.2 Bounding syntactic resources

We present our decidability/complexity/expressivity results by considering different fragments. We restrict the syntactic resources in at least two ways, seldom simultaneously.

1. Bounding the number of variables is quite natural (see Lemma 1), similarly to restricting the number of propositional variables in modal and temporal logics, see e.g. (Halpern 1995) and (Spaan 1993, Thm. 5.4.6).
2. Bounding the term depth (i.e., the values n, m) in expressions of the form $\mathsf{X}^n\mathtt{x} \sim \mathsf{X}^m\mathtt{y}$ or $\mathsf{X}^n\mathtt{x} = d$, or in assignment operators $\langle \mathtt{x}{:=}\mathsf{X}^n\mathtt{y} \rangle$ allows us to compare or assign values that are very close to the current position. Typically, the renaming technique can be used to reduce term depth n to a value in $\{-1, 0, +1\}$, at the cost of introducing new auxiliary variables. If $n \in \{-1, 0, +1\}$, then we say that $\langle \mathtt{x}{:=}\mathsf{X}^n\mathtt{y} \rangle$ is an **adjacent assignment**.

It is easy to eliminate assignments of the form $\langle \mathtt{x}{:=}d \rangle$ by adding a new variable $\mathtt{y}_d$, to require the satisfaction of $\mathsf{G}(\mathtt{y}_d = d)$ and to replace $\langle \mathtt{x}{:=}d \rangle$ by $\langle \mathtt{x}{:=}\mathtt{y}_d \rangle$. In the sequel, whenever the number of variables is not bounded, we do not need to make use of assignments of the form $\langle \mathtt{x}{:=}d \rangle$.

The following results state that we can restrict the number of variables, without affecting complexity bounds.

Lemma 1. *There is a logspace reduction from the satisfiability problem for* PLTL-A$(\mathbb{D})$ *to its restriction to formulae with one variable only.*

The proof follows a standard pattern: we encode n variables into a sequence of length n with only one variable. Thus, each position in a model for an arbitrary formula is encoded as a sequence. The main challenge is to uniquely identify where each sequence starts. This is done by "marking" each of such sequences with a prefix $d^{n+2}d'$, where d^{n+2} stands for $n+2$ times the value d, and $d' \neq d$. The encoding of a valuation at a particular position has the following shape:

$$d^{n+2}\ d'\ \sigma(i)(\mathtt{x}_1)\ \sigma(i)(\mathtt{x}_2)\ \ldots\ \sigma(i)(\mathtt{x}_n).$$

Thus, the only way of finding a sequence of $n+2$ consecutive values, followed by a value that is different from them, is at the beginning of each sequence. It is worth noticing that for this encoding, we only need the concrete domain $\mathbb{D}$ to have at least two values, slightly improving the encoding from (Demri, Lazić, and Nowak 2007, Prop. 4).

Example 2. *Let us consider the following encoding, and show how the translation works. Suppose we have three variables* $\mathtt{x}_1, \mathtt{x}_2, \mathtt{x}_3$*, i.e.,* $n = 3$*. Thus,* σ *is encoded as:*

$$\mathbf{d^5}\mathbf{d'}\boxed{d_1^0}d_2^0d_3^0\mathbf{d^5}\mathbf{d'}d_1^1d_2^1d_3^1\underset{\uparrow}{\mathbf{d^5}}\mathbf{d'}d_1^2\boxed{d_2^2}d_3^2\ldots$$

where the leftmost subsequence corresponds to position 0*,* d_j^i *stands for* $\sigma(i)(\mathtt{x}_j)$ *and* $\mathbf{d^5}$ *is a sequence made of five copies of* $\mathbf{d}$*. At each position, the single variable* $\mathtt{x}$ *stores the respective value* d_j^i*. Suppose we want to perform the update* $\langle \mathtt{x}_2{:=}\mathsf{X}^{-2}\mathtt{x}_1 \rangle$ *at position* 2*. This position is marked in the encoding with an* $\uparrow$ *below. The encoding of the assignment results as* $\mathsf{X}^7\langle \mathtt{x}{:=}\mathsf{X}^{-19}\mathtt{x} \rangle$*. It is easy to check that the involved values are those inside a box.*

Below, we consider another syntactic restriction by admitting only adjacent assignments but at the cost of allowing two variables (instead of a single one earlier). The second variable is used to carry a value over adjacent assignments.

Lemma 2. *There is a logspace reduction from the satisfiability problem for* PLTL-A$(\mathbb{D})$ *to its restriction to formulae at most two variables and adjacent assignments only.*

Observe that Lemmas 1 and 2 also hold for any concrete domain $\mathbb{D}$ equipped with the equality relation and not necessarily for domains of the form $(\mathbb{D}, =, <, (=_d)_{d\in\mathbb{D}})$. Indeed, the proofs can be slightly adapted.

To conclude this section, after having manipulated PLTL-A($\mathbb{D}$) formulae and the satisfaction relation for PLTL-A($\mathbb{D}$), it should be clear that the assignments of the form $\langle \mathtt{x}{:=}\mathsf{X}^n\mathtt{y}\rangle$ are only performed locally and the modified local value for $\mathtt{x}$ cannot be accessed directly *all over the models*, unlike the effect of the freeze operator, see e.g. (Figueira and Segoufin 2009).

3 Satisfiability Problem for PLTL-A($\{0,1\}$)

In this section, we investigate the decidability/complexity status of the satisfiability problem for the logic PLTL-A($\{0,1\}$) whereas Sec. 4 is devoted to the more general case PLTL-A($\mathbb{D}$) with an arbitrary concrete domain $\mathbb{D}$. We provide elementary reductions between PLTL-A($\{0,1\}$) and the first-order logic over ω-words, known to admit a Tower-complete satisfiability problem and to be expressively equivalent to the linear-time temporal logic PLTL (Kamp 1968). The class Tower introduced in (Schmitz 2016) is closed under elementary reductions and Tower-hardness captures non-elementarity. In order to stick to the notations about PLTL, first we present the logic PLTL$^{:=}$ defined as a syntactic variant of the logic PLTL-A($\{0,1\}$) in which we directly manipulate propositional variables and where the assignment operator updates their truth values. Hence, all the results presented below about PLTL$^{:=}$ can be easily rephrased for PLTL-A($\{0,1\}$).

Introducing PLTL$^{:=}$. Let Prop be a countably infinite set of propositional symbols. The set of PLTL$^{:=}$ formulae is defined by the following grammar:

$$\varphi,\psi ::= \mathtt{p} \mid \varphi\vee\psi \mid \neg\varphi \mid \mathsf{X}\varphi \mid \mathsf{X}^{-1}\varphi \mid \varphi\mathsf{U}\psi \mid \varphi\mathsf{S}\psi \mid \langle \mathtt{p}{:=}\mathsf{X}^n\mathtt{q}\rangle\varphi \mid \langle \mathtt{p}{:=}\top\rangle\varphi \mid \langle \mathtt{p}{:=}\bot\rangle\varphi,$$

where $\mathtt{p},\mathtt{q}\in\mathsf{Prop}$ and $n\in\mathbb{Z}$. A **model for PLTL$^{:=}$** is a map $\sigma:\mathbb{N}\to(\mathsf{Prop}\to\{0,1\})$. Let σ be a model and $i\geqslant 0$. The **satisfaction relation** $\Vdash$ between σ and formulae is inductively defined as follows:

$$\begin{array}{lcl}
\sigma,i\Vdash\mathtt{p} & \overset{\text{def}}{\Leftrightarrow} & \sigma(i)(\mathtt{p})=1\\
\sigma,i\Vdash\langle \mathtt{p}{:=}\top\rangle\varphi & \overset{\text{def}}{\Leftrightarrow} & \sigma[\mathtt{p}\mapsto_i 1],i\Vdash\varphi\\
\sigma,i\Vdash\langle \mathtt{p}{:=}\bot\rangle\varphi & \overset{\text{def}}{\Leftrightarrow} & \sigma[\mathtt{p}\mapsto_i 0],i\Vdash\varphi\\
\sigma,i\Vdash\langle \mathtt{p}{:=}\mathsf{X}^n\mathtt{q}\rangle\varphi & \overset{\text{def}}{\Leftrightarrow} & i+n\geqslant 0 \text{ and } \sigma[\mathtt{p}\mapsto_i\sigma(i+n)(\mathtt{q})],i\Vdash\varphi.
\end{array}$$

3.1 Hardness and first-order logic over ω-words

In order to analyse the decidability/complexity status of PLTL$^{:=}$, we need to introduce first-order logic over ω-words. Then, we will rely on results from (Stockmeyer 1974).

First-order logic FO(ω). We use a signature containing Prop (used as a countable set of unary predicate symbols), two binary symbols $<$ and $=$ (standing for 'less than' and 'equal', respectively) and a function $+1$ standing for successor function. Models in FO(ω) are models as for PLTL$^{:=}$.

The syntax of FO(ω) is given by the following grammar:

$$\varphi,\psi ::= \mathtt{x}=\mathtt{y} \mid \mathtt{x}<\mathtt{y} \mid \mathtt{x}=\mathtt{y}+1 \mid \neg\varphi \mid \varphi\vee\psi \mid \mathtt{p}(\mathtt{x}) \mid \exists\mathtt{x}\varphi.$$

Let σ be a model, and let $g:\mathsf{Var}\mapsto\mathbb{N}$ be a function that assigns to each variable a position (called **environment**), the **satisfaction relation** $\Vdash$ for FO(ω) is defined as follows (Boolean cases are omitted):

$$\begin{array}{lcl}
\sigma\Vdash_g \mathtt{x}\star\mathtt{y} & \overset{\text{def}}{\Leftrightarrow} & g(\mathtt{x})\star g(\mathtt{y}) \quad (\text{for } \star\in\{=,<\})\\
\sigma\Vdash_g \mathtt{x}=\mathtt{y}+1 & \overset{\text{def}}{\Leftrightarrow} & g(\mathtt{x})=g(\mathtt{y})+1\\
\sigma\Vdash_g \mathtt{p}(\mathtt{x}) & \overset{\text{def}}{\Leftrightarrow} & \sigma(g(\mathtt{x}))(\mathtt{p})=1\\
\sigma\Vdash_g \exists\mathtt{x}\varphi & \overset{\text{def}}{\Leftrightarrow} & \text{there is } k\geqslant 0 \text{ such that } \sigma\Vdash_{g'}\varphi,
\end{array}$$

where g' is exactly as g except that $g'(\mathtt{x})=k$. An FO(ω) formula φ is **satisfiable** iff there are a model σ and a function g such that $\sigma\Vdash_g\varphi$.

It is shown in (Stockmeyer 1974, Thm. 5.2) that the nonemptiness problem for star-free regular expressions can be reduced in polynomial time to the satisfiability problem for FO(ω). By (Schmitz 2016, Sec. 3.1), the equivalence problem SFEq for star-free expressions (that can be reduced to the nonemptiness problem) is Tower-complete. Consequently, satisfiability problem for FO(ω) is Tower-hard. Moreover, FO(ω) is a fragment of the monadic second-order logic MSO over ω-words and MSO formulae can be turned into equi-expressive Büchi automata (Büchi 1962) with an effective automaton construction. This entails a Tower-membership for MSO satisfiability, and thus for FO(ω). Tower-hardness of FO(ω) is used to show Tower-hardness of the satisfiability problem of PLTL$^{:=}$.

Lemma 3. *There is a logspace reduction from the satisfiability problem for FO(ω) to the one for PLTL$^{:=}$.*

Proof. Let φ be an FO(ω) sentence built over the unary predicates $\mathtt{p}_1,\dots,\mathtt{p}_\beta$ and the individual variables $\mathtt{x}_1,\dots,\mathtt{x}_\alpha$. Without any loss of generality, we can assume that distinct quantifier occurrences use distinct variables. We define a translation map $\mathbf{t}$ into PLTL$^{:=}$ using the auxiliary propositional variables $\mathtt{q}_1,\dots,\mathtt{q}_\alpha$ such that φ is FO(ω)-satisfiable iff $\mathsf{G}(\bigwedge_{j=1}^{\alpha}\neg\mathtt{q}_j)\wedge\mathbf{t}(\varphi)$ is PLTL$^{:=}$-satisfiable.

The role of each propositional variable $\mathtt{q}_j$ is to hold true at most at one position k, in order to encode the fact that the variable $\mathtt{x}_j$ is interpreted by the position k. Initially, $\mathtt{q}_j$ does not hold anywhere, which means that no value is assigned to the variable $\mathtt{x}_j$. The translation $\mathbf{t}$ is homomorphic for Boolean connectives and satisfies the clauses below.

- $\mathbf{t}(\exists\mathtt{x}_j\,\psi)\overset{\text{def}}{=}\mathsf{E}\langle\mathtt{q}_j{:=}\top\rangle\,\mathbf{t}(\psi)$ ($\mathsf{E}\chi$ shortcut for $\mathsf{F}\chi\vee\mathsf{P}\chi$),
- $\mathbf{t}(\mathtt{x}_i=\mathtt{x}_j)\overset{\text{def}}{=}\mathsf{E}(\mathtt{q}_i\wedge\mathtt{q}_j)$; $\mathbf{t}(\mathtt{x}_j=\mathtt{x}_i+1)\overset{\text{def}}{=}\mathsf{E}(\mathtt{q}_i\wedge\mathsf{X}\mathtt{q}_j)$.
- $\mathbf{t}(\mathtt{x}_i<\mathtt{x}_j)\overset{\text{def}}{=}\mathsf{E}(\mathtt{q}_i\wedge\mathsf{XF}\mathtt{q}_j)$; $\mathbf{t}(\mathtt{p}_i(\mathtt{x}_j))\overset{\text{def}}{=}\mathsf{E}(\mathtt{p}_i\wedge\mathtt{q}_j)$.

In order to prove the correctness of the reduction, we introduce a relation $\sigma,g\sim\sigma'$ between an FO(ω)-model σ and an environment $g:\{\mathtt{x}_1,\dots,\mathtt{x}_\alpha\}\rightharpoonup\mathbb{N}$ (partial function) and an PLTL$^{:=}$ model $\sigma':\mathbb{N}\to(\{\mathtt{p}_1,\dots,\mathtt{p}_\beta,\mathtt{q}_1,\dots,\mathtt{q}_\alpha\}\to\{0,1\})$ satisfying the following clauses.

- For all $i\in\mathbb{N}$ and $j\in[1,\beta]$, $\sigma(i)(\mathtt{p}_j)=\sigma'(i)(\mathtt{p}_j)$.
- Let $j\in[1,\alpha]$. If the map g is defined on $\mathtt{x}_j$, then $\sigma'(g(\mathtt{x}_j))(\mathtt{q}_j)=1$ and for all $i\neq g(\mathtt{x}_j)$, $\sigma'(i)(\mathtt{q}_j)=0$. Otherwise, for all $i\in\mathbb{N}$, $\sigma'(i)(\mathtt{q}_j)=0$.

The following properties can be easily established and are useful to complete the proof.

(P1) Given σ and $g : \{\mathtt{x}_1, \ldots, \mathtt{x}_\alpha\} \rightharpoonup \mathbb{N}$, there is a unique model σ' such that $\sigma, g \sim \sigma'$.

(P2) Given $\sigma' : \mathbb{N} \to (\{\mathtt{p}_1, \ldots, \mathtt{p}_\beta, \mathtt{q}_1, \ldots, \mathtt{q}_\alpha\} \to \{0,1\})$ such that for all $j \in [1, \alpha]$, there is at most one position $i \in \mathbb{N}$ such that $\sigma'(i)(\mathtt{q}_j) = 1$, there are a unique model σ and environment g such that $\sigma, g \sim \sigma'$.

By structural induction, one can show that if $\sigma, g \sim \sigma'$, then for all subformulae ψ of φ such that g is defined for the free variables in ψ, we have $\sigma \Vdash_g \psi$ iff (for all $i \in \mathbb{N}$, we have $\sigma', i \Vdash \mathbf{t}(\psi)$). □

Notice that the reduction in the proof of Lemma 3 is logspace because the size of formulae is assumed to be the cardinality of its set of subformulae. Any other reasonable definitions for sizes would lead anyhow to elementary reductions, which is sufficient to get Tower-hardness. By using Lemmas 1 and 2, as well as Lemma 3 and the Tower-hard of $\mathsf{FO}(\omega)$, we get the following results.

Lemma 4. *The satisfiability problem of $\mathsf{PLTL}^{:=}$ restricted to formulae with a single propositional variable (resp. with at most two propositional variables and adjacent assignments only) is* Tower*-hard.*

3.2 Introduction to symbolic assignments

In order to show that the satisfiability problem for $\mathsf{PLTL}^{:=}$ is in Tower, we define a faithful and polynomial-space reduction to $\mathsf{FO}(\omega)$ (see Sec. 3.3).

First, let us identify a fragment of $\mathsf{PLTL}^{:=}$ to which the satisfiability problem for $\mathsf{PLTL}^{:=}$ can be reduced in logspace, and from which the forthcoming translation to $\mathsf{FO}(\omega)$ becomes smoother. The **flat fragment** of $\mathsf{PLTL}^{:=}$ is defined as the fragment of $\mathsf{PLTL}^{:=}$ in which the only authorised modalities of the form $\langle \mathtt{p}{:=}\mathsf{X}^n\mathtt{q}\rangle$ satisfy $n = 0$, i.e., no strict past or future truth values are involved.

Lemma 5. *There is a logspace reduction from the satisfiability problem for $\mathsf{PLTL}^{:=}$ to the one for its flat fragment.*

Proof. (sketch) Let φ be a $\mathsf{PLTL}^{:=}$ formula and $M \in \mathbb{N}$ be the minimal value such that for all modalities $\langle \mathtt{p}{:=}\mathsf{X}^n\mathtt{q}\rangle$ occurring in φ, we have $|n| \leqslant M$. Let $\mathtt{p}_1, \ldots, \mathtt{p}_N$ be the propositional variables occurring in φ. To define the target formula in the flat fragment, we introduce the variables $\mathtt{p}_j^i$ with $i \in [-M, +M]$ so that $\mathtt{p}_j^i$ plays the role of the term $\mathsf{X}^i\mathtt{p}_j$. The formula $\varphi_{N,M}$ defined below specifies unambiguously what is expected from the new propositional variables.

$$\mathsf{G}\Big(\bigwedge_{i\in[0,M\text{-}1],j\in[1,N]} (\mathtt{p}_j^{-i} \Leftrightarrow \mathsf{X}\mathtt{p}_j^{-(i+1)} \wedge \mathtt{p}_j^{i+1} \Leftrightarrow \mathsf{X}\mathtt{p}_j^i) \wedge \bigwedge_{j\in[1,N]} \mathtt{p}_j \Leftrightarrow \mathtt{p}_j^0\Big).$$

Furthermore, we define a translation map $\mathbf{t}$ that is homomorphic for Boolean and temporal connectives, for modalities of the form $\langle \mathtt{p}_j{:=}\top\rangle$ and $\langle \mathtt{p}_j{:=}\bot\rangle$, and is equal to identity for propositional variables. The new variables that are part of the renaming techniques play their full role for the translation of formulae whose outermost connective is of the form $\langle \mathtt{p}_j{:=}\mathsf{X}^n\mathtt{p}_k\rangle$. However, after performing local changes, we need to maintain the satisfaction of $\varphi_{N,M}$ (with the new value), which causes some complications in the definition of the map $\mathbf{t}$. Typically, changing the value of $\mathtt{p}_j$ at position i, requires to update the variable $\mathtt{p}_j^m$ at position $i + m$, for all $m \in [-M, M]$ such that $i + m \geqslant 0$. Below, $n \in [-M, M]$.

$$\mathbf{t}(\langle \mathtt{p}_j{:=}\mathsf{X}^n\mathtt{p}_k\rangle\,\psi) \stackrel{\text{def}}{=} \mathsf{X}^n\top \wedge \langle \mathtt{p}_j{:=}\mathtt{p}_k^n\rangle\,((\mathtt{p}_j \Rightarrow \psi_\top) \wedge (\neg\mathtt{p}_j \Rightarrow \psi_\bot)),$$

where the formulae $\psi_\top$ and $\psi_\bot$ are defined below. Given a truth value $b \in \{\bot, \top\}$ and $\gamma \in [1, M]$, we write $\mathsf{O}^b(\gamma, \psi_1, \psi_2)$ to denote the formula $\mathsf{X}^{-1}\top \Rightarrow \mathsf{X}^{-1}\langle \mathtt{p}_j^\gamma{:=}b\rangle\,\psi_2 \wedge \neg\mathsf{X}^{-1}\top \Rightarrow \mathsf{X}^{\gamma-1}\psi_1$.

$$\psi_b \stackrel{\text{def}}{=} \langle \mathtt{p}_j^0{:=}b\rangle\,(\mathsf{X}\langle \mathtt{p}_j^{-1}{:=}b\rangle) \cdots (\mathsf{X}\langle \mathtt{p}_j^{-M}{:=}b\rangle)\mathsf{X}^{-M}$$
$$\mathsf{O}^b(1, \mathbf{t}(\psi), \mathsf{O}^b(2, \mathbf{t}(\psi), \mathsf{O}^b(3, \mathbf{t}(\psi), \ldots$$
$$\mathsf{O}^b(M, \mathbf{t}(\psi), \mathsf{X}^M\,\mathbf{t}(\psi))\ldots))).$$

It is easy to check that $\varphi_{N,M} \wedge \mathbf{t}(\varphi)$ belongs to the flat fragment of $\mathsf{PLTL}^{:=}$ and assuming that the size of a formula is its number of subformulae, it can be computed in logspace in the size of φ. One can show that φ is satisfiable iff $\varphi_{N,M} \wedge \mathbf{t}(\varphi)$ is satisfiable. □

For the translation into $\mathsf{FO}(\omega)$, we can assume that the source formula belongs to the flat fragment of $\mathsf{PLTL}^{:=}$. Furthermore, before defining the translation, we introduce auxiliary notions such as symbolic assignments and symbolic contexts. Such syntactic objects are designed so that an argument of the translation map (to be defined) contains a symbolic context that corresponds to the sequence of symbolic assignments that have been performed so far.

A **symbolic assignment** is defined as an expression of the form $(\mathtt{p}, e, \mathtt{y})$, where $\mathtt{p}$ is a propositional variable, $\mathtt{y}$ is an individual variable, and e is either a truth constant or a propositional variable $\mathtt{q}$. By way of example, $(\mathtt{p}, \mathtt{q}, \mathtt{y})$ encodes the process of updating the truth value of the propositional variable $\mathtt{p}$ at the position $\mathtt{y}$ with the truth value of the propositional variable $\mathtt{q}$ at the position $\mathtt{y}$. For such a statement to be formally correct, we still need an environment g to interpret the variable $\mathtt{y}$ so that we can really specify a position in $\mathbb{N}$. Given a sequence $\mathcal{S}$ of symbolic assignments, written $(\mathtt{p}_{i_1}, e_1, \mathtt{y}_1), \ldots, (\mathtt{p}_{i_N}, e_N, \mathtt{y}_N)$, we write $\mathcal{S}[j, j']$ to denote its subsequence $(\mathtt{p}_{i_j}, e_j, \mathtt{y}_j), \ldots, (\mathtt{p}_{i_{j'}}, e_{j'}, \mathtt{y}_{j'})$. Such a sequence $\mathcal{S}$ is called a **symbolic context**.

Given a symbolic context $\mathcal{S}$, to be able to apply the sequence of symbolic assignments from $\mathcal{S}$ to a given model σ, we need an environment g to interpret the individual variables occurring in $\mathcal{S}$. That is why we define below the model $\mathit{update}(\sigma, \mathcal{S}, g)$ that is intended to be the model σ on which the local changes from the symbolic context $\mathcal{S}$ are performed but under the environment g. Assuming that $\mathcal{S}$ is of the form $(\mathtt{p}_{i_1}, e_1, \mathtt{y}_1), \ldots, (\mathtt{p}_{i_N}, e_N, \mathtt{y}_N)$, the model $\mathit{update}(\sigma, \mathcal{S}, g)$ is defined according to the clauses below.

- $\mathit{update}(\sigma, \varepsilon, g) \stackrel{\text{def}}{=} \sigma$,
- if $N > 0$, then $\mathit{update}(\sigma, \mathcal{S}, g) \stackrel{\text{def}}{=}$

  $$\mathit{update}(\sigma, \mathcal{S}[1, N-1], g)[\mathtt{p}_{i_N} \mapsto_{g(\mathtt{y}_N)} V_N],$$

  where $V_N = \mathit{update}(\sigma, \mathcal{S}[1, N-1], g)(g(\mathtt{y}_N))(\mathtt{q})$ if $e_N = \mathtt{q}$, $V_N = 1$ if $e_N = \top$, and $V_N = 0$ if $e_N = \bot$.

The last preliminary property that we need to establish is related to the value $update(\sigma, \mathcal{S}, g)(i)(\mathtt{p})$ for some position i and some propositional variable $\mathtt{p}$. Indeed, Lemma 6 below shall be essential to translate propositional variables.

Lemma 6. *Let σ be a model, g be an environment, $\mathcal{S}$ be a symbolic context $(\mathtt{p}_{i_1}, e_1, \mathtt{y}_1), \ldots, (\mathtt{p}_{i_N}, e_N, \mathtt{y}_N)$, $i \in \mathbb{N}$ and $\mathtt{p} \in \mathsf{Prop}$. The value $update(\sigma, \mathcal{S}, g)(i)(\mathtt{p})$ is obtained among the three disjoint cases below.*

(I) *If there is no k such that $\mathtt{p} = \mathtt{p}_{i_k}$, then $update(\sigma, \mathcal{S}, g)(i)(\mathtt{p}) = \sigma(i)(\mathtt{p})$.*

(II) *If (I) does not hold and for all k such that $\mathtt{p} = \mathtt{p}_{i_k}$, we have $g(\mathtt{y}_k) \neq i$, then $update(\sigma, \mathcal{S}, g)(i)(\mathtt{p}) = \sigma(i)(\mathtt{p})$.*

(III) *If both (I) and (II) do not hold and $k = \max\{k' \mid g(\mathtt{y}_{k'}) = i,\ \mathtt{p}_{i_{k'}} = \mathtt{p}\}$, then $update(\sigma, \mathcal{S}, g)(i)(\mathtt{p}) = update(\sigma, \mathcal{S}[1, k{-}1], g)(i)(\mathtt{q})$ with $e_k = \mathtt{q}$. If e_k takes the value $\bot$ or $\top$, then $update(\sigma, \mathcal{S}, g)(i)(\mathtt{p})$ takes the value 0 or 1, respectively.*

3.3 The translation map

We define a translation map $\mathbf{t}$ from the flat fragment of $\mathsf{PLTL}^{:=}$ to $\mathsf{FO}(\omega)$ that generalises the translation from PLTL into first-order logic (Kamp 1968). Instead of having only two arguments, namely the formula φ to be translated and an individual variable $\mathtt{y}$ interpreted as the position on which the formula φ is evaluated, we add a third argument that encodes the succession of assignments, namely a symbolic context. The new cases are those for propositional variables and for assignments.

- $\mathbf{t}(\psi \vee \psi', \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} \mathbf{t}(\psi, \mathtt{y}, \mathcal{S}) \vee \mathbf{t}(\psi', \mathtt{y}, \mathcal{S})$. The translation for other Boolean cases is similar. Moreover, $\mathbf{t}(\top, \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} \top$, $\mathbf{t}(\bot, \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} \bot$ and $\mathbf{t}(\psi \mathsf{S} \psi', \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} \exists \mathtt{y}'\ (\mathtt{y}' \leqslant \mathtt{y}) \wedge \mathbf{t}(\psi', \mathtt{y}', \mathcal{S}) \wedge (\forall \mathtt{y}''(\mathtt{y}' < \mathtt{y}'' \leqslant \mathtt{y}) \Rightarrow \mathbf{t}(\psi, \mathtt{y}'', \mathcal{S}))$. The translation of formulae with another outermost temporal connective, is defined similarly based on the internalisation of PLTL semantics into $\mathsf{FO}(\omega)$.
- The new cases are handled now. To start with, $\mathbf{t}(\langle \mathtt{p}{:=}\mathtt{q} \rangle \psi, \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} \mathbf{t}(\psi, \mathtt{y}, \mathcal{S} \cdot (\mathtt{p}, \mathtt{q}, \mathtt{y}))$ and for $B \in \{\bot, \top\}$, $\mathbf{t}(\langle \mathtt{p}{:=}B \rangle \psi, \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} \mathbf{t}(\psi, \mathtt{y}, \mathcal{S} \cdot (\mathtt{p}, B, \mathtt{y}))$.
- If there is no j such that $\mathtt{p} = \mathtt{p}_{i_j}$ from $\mathcal{S}$, then $\mathbf{t}(\mathtt{p}, \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} \mathtt{p}(\mathtt{y})$ (which is the usual definition for PLTL, say with $\mathcal{S} = \varepsilon$). This corresponds to case (I) in Lemma 6. Otherwise,

$$\mathbf{t}(\mathtt{p}, \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} \Big(\bigwedge_{j \text{ s.t. } \mathtt{p}_{i_j} = \mathtt{p}} (\mathtt{y}_j \neq \mathtt{y}) \wedge \mathtt{p}(\mathtt{y}) \Big) \vee$$

$$\bigvee_{j \text{ s.t. } \mathtt{p}_{i_j} = \mathtt{p}} \Big(\bigwedge_{j' \text{ s.t. } \mathtt{p}_{i_{j'}} = \mathtt{p} \text{ and } j' > j} (\mathtt{y}_{j'} \neq \mathtt{y}) \Big) \wedge (\mathtt{y}_j = \mathtt{y}) \wedge$$

$$\mathbf{t}(e_j, \mathtt{y}_j, \mathcal{S}[1, j-1]).$$

  The first (resp. second) line corresponds to case (II) (resp. (III)), concluding the definition of $\mathbf{t}$.

It is also possible to use shortcuts, for instance to translate E-formulae with the clause $\mathbf{t}(\mathsf{E}\psi, \mathtt{y}, \mathcal{S}) = \exists \mathtt{y}'\ \mathbf{t}(\psi, \mathtt{y}', \mathcal{S})$. By way of example, the translation of $\mathsf{E}\langle \mathtt{p}{:=}\mathtt{q} \rangle \mathsf{E}\langle \mathtt{r}{:=}\mathtt{p} \rangle \mathtt{r}$ is provided below (after a few propositional simplifications):

$$\exists \mathtt{y}_1\ \exists \mathtt{y}_2\ \big((\mathtt{y}_2 = \mathtt{y}_1) \wedge \mathtt{q}(\mathtt{y}_1)\big) \vee \big((\mathtt{y}_2 \neq \mathtt{y}_1) \wedge \mathtt{p}(\mathtt{y}_2)\big).$$

Lemma 7. *φ is $\mathsf{PLTL}^{:=}$ satisfiable iff $\exists \mathtt{y}\ (\neg \exists \mathtt{y}'\ \mathtt{y}' < \mathtt{y}) \wedge \mathbf{t}(\varphi, \mathtt{y}, \varepsilon)$ is $\mathsf{FO}(\omega)$ satisfiable.*

In Lemma 7, we do not *only* use $\mathbf{t}(\varphi, \mathtt{y}, \varepsilon)$ as a target formula in $\mathsf{FO}(\omega)$ because satisfiability of $\mathsf{PLTL}^{:=}$ formulae is defined from the origin position. Then, we get:

Lemma 8. *There is a polynomial-space reduction from the satisfiability problem for $\mathsf{PLTL}^{:=}$ to the one for $\mathsf{FO}(\omega)$.*

We designed a reduction to $\mathsf{FO}(\omega)$ leading to the decidability of the satisfiability problem for $\mathsf{PLTL}^{:=}$. We also get Tower-membership since the map $\mathbf{t}$ induces a polynomial-space reduction to the satisfiability problem for $\mathsf{FO}(\omega)$.

Variants of symbolic contexts. The idea of using a context made of the symbolic representation of a finite set of assignments to perform the translation into $\mathsf{FO}(\omega)$ is reminiscent of the translation from sabotage modal logic and from other relation-changing logics into first-order logic, see e.g. (Aucher, van Benthem, and Grossi 2018, Sec. 2.3) and (Areces, Fervari, and Hoffmann 2015, Sec. 3.2). There, a context made of the symbolic representation of a finite set of edges is carried along the translation. Indeed, the two types of context encode different pieces of information but in both cases, there is some record of a finite amount of information about the modifications made on the original model. As far as assignments are concerned, the ordering is important; that is why sequences of the symbolic representation of assignments are involved. It is also worth noting that in (Belardinelli et al. 2023, Sec. 3), a program-epistemic logic admitting assignments is translated into FO but it is based on principles different from ours, though it obviously shares the idea of internalising the semantics of the source logic.

Extension to arbitrary formula assignments. In $\mathsf{PLTL}^{:=}$, it is possible to assign constant truth values or the values of propositional variables. However, we may wish to allow expressions of the form $\langle \mathtt{p}{:=}\varphi \rangle \psi$, where φ is an arbitrary formula of $\mathsf{PLTL}^{:=}$ (as done globally in e.g. (van Ditmarsch, Herzig, and de Lima 2012; Galimullin et al. 2025)). The formula $\langle \mathtt{p}{:=}\varphi \rangle \psi$ can be internalised in $\mathsf{PLTL}^{:=}$ by:

$$\varphi \Rightarrow \langle \mathtt{p}{:=}\top \rangle \psi \wedge \neg\varphi \Rightarrow \langle \mathtt{p}{:=}\bot \rangle \psi).$$

Thus, we directly obtain:

Theorem 1. *The satisfiability problem for $\mathsf{PLTL}^{:=}$ extended with arbitrary assignments is Tower-complete.*

As a consequence, the satisfiability problem for $\mathsf{PLTL\text{-}A}(\{0,1\})$ is also Tower-complete. It is worth noting that we would get a similar complexity with the model-checking problem. Indeed, given a set of propositional variables $\mathtt{p}_1, \ldots, \mathtt{p}_n$, the set of models for $\mathsf{PLTL}^{:=}$ can be generated by some one-state transition system with 2^n edges labelled by assignments on $\mathtt{p}_1, \ldots, \mathtt{p}_n$. Consequently, the existential model-checking problem for $\mathsf{FO}(\omega)$ is also Tower-hard (by an exponential-time reduction from the satisfiability problem). Using the encoding of Lemma 3, we conclude that the existential model-checking problem for $\mathsf{PLTL\text{-}A}$ is also Tower-hard. For concrete domains handled in Section 4, we can get a similar conclusion.

4 How to Decide Satisfiability for PLTL-A($\mathbb{D}$)

In this section, we consider the satisfiability problem for the linear-time temporal logic PLTL-A($\mathbb{D}$) with $\mathbb{D} \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}\}$. To do so, we rely on the proof methods from Sec. 3 dedicated to PLTL-A($\{0,1\}$) though we generalise them in several directions. Recall that in the case of PLTL-A($\mathbb{D}$), in the formula $\langle \mathtt{x}{:=}\mathsf{X}\mathtt{x}' \rangle \varphi$, the variable $\mathtt{x}$ takes the next value of $\mathtt{x}'$ that belongs to a potentially infinite set $\mathbb{D}$. The reduction from PLTL-A($\{0,1\}$) to FO(ω) in Sec. 3 can be extended to PLTL-A($\mathbb{D}$). However, we need a target logic interpreted over ω-sequences of tuples in $\mathbb{D}^\beta$ and therefore it is necessary to consider the appropriate decidable logic on multi-attributed data words if it exists. We know that first-order logic over multi-attributed data words with two data values per position is undecidable (even restricted to two variables), as well as over data words with one data value per position and three quantified variables (Bojańczyk et al. 2011) (see also other logics over multi-attributed data words in (Decker et al. 2014; Bollig, Sangnier, and Stietel 2024)).

Additionally, the satisfiability problem for FO2 with one datum per position and data equality only is shown equivalent to the reachability problem for Petri nets (Bojańczyk et al. 2011). This problem is shown Ack-complete (Leroux and Schmitz 2019; Czerwinski and Orlikowski 2021; Leroux 2021). To aim for decidability here, it is necessary to devise appropriate syntactic restrictions. This is precisely what we perform in Sec. 4.1 to design the logic $\mathsf{FO}^{\mathrm{loc}}(\beta)$.

To start with, we establish Tower-hardness by using the Tower-hardness of PLTL-A($\{0,1\}$), and requiring that every data value at every position is either 0 or 1.

Theorem 2. *For all $\mathbb{D} \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}\}$, the satisfiability problem for PLTL-A($\mathbb{D}$) is* Tower*-hard.*

In order to prove that PLTL-A($\mathbb{D}$) is decidable, we introduce a new first-order logic interpreted on multi-attributed data words whose satisfiability problem is shown decidable.

4.1 A new logic on multi-attributed data words

Given $\beta \geqslant 1$, we define the first-order logic $\mathsf{FO}^{\mathrm{loc}}(\beta)$ interpreted on multi-attributed data words in $(\mathbb{D}^\beta)^\omega$. Observe that our multi-attributed data words do not use any additional finite alphabet because this can be simulated by increasing the dimension β and using equality constraints. The dimension β is intended to capture the number of variables in the PLTL-A($\mathbb{D}$) formula to be translated.

Quantifications in $\mathsf{FO}^{\mathrm{loc}}(\beta)$ range over positions of the infinite multi-attributed data word as in FO(ω). The language for $\mathsf{FO}^{\mathrm{loc}}(\beta)$ includes comparison predicates for positions $=$ and $<$, as well as the successor relation $+1$. To access the β data values at a given position, we introduce the unary symbols $f_1, \ldots, f_\beta$ (as concrete features in description logics with concrete domains, see e.g. (Borgwardt, Bortoli, and Koopmann 2024)). Recall that to get decidability, syntactic restrictions are necessary. This is precisely what we perform below to define $\mathsf{FO}^{\mathrm{loc}}(\beta)$ by requiring that data constraints of the form $f_i(\mathtt{x}+k_1) \simeq f_j(\mathtt{x}'+k_2)$ and $f_i(\mathtt{x}+k_1) \prec f_j(\mathtt{x}'+k_2)$ (with shifts $k_1, k_2 \in \mathbb{Z}$) are authorised *only if* $\mathtt{x}$ is equal to $\mathtt{x}'$ syntactically.

The syntax of $\mathsf{FO}^{\mathrm{loc}}(\beta)$ is defined by the grammar:

$$\begin{aligned}\varphi, \psi ::= {} & \mathtt{x} = \mathtt{y} \mid \mathtt{x} < \mathtt{y} \mid \mathtt{x} = \mathtt{y} + 1 \mid \neg\varphi \mid \varphi \vee \psi \\ & \mid f_i(\mathtt{x}+k_1) \simeq f_j(\mathtt{x}+k_2) \mid f_i(\mathtt{x}+k_1) \simeq d \\ & \mid f_i(\mathtt{x}+k_1) \prec f_j(\mathtt{x}+k_2) \mid \exists \mathtt{x} \varphi,\end{aligned}$$

with $\mathtt{x}, \mathtt{y} \in \mathsf{Var}$, $k_1, k_2 \in \mathbb{Z}$, $i, j \in [1, \beta]$ and $d \in \mathbb{D}$. Models are of the form $(\mathbb{D}^\beta)^\omega$. Given $w \in (\mathbb{D}^\beta)^\omega$ and $g : \mathsf{Var} \to \mathbb{N}$, the relation $\Vdash$ is defined as follows (Boolean cases omitted).

$$\begin{aligned}
w \Vdash_g \mathtt{x} \star \mathtt{y} \quad & \overset{\text{def}}{\Leftrightarrow} \quad g(\mathtt{x}) \star g(\mathtt{y}) \quad (\star \in \{=, <\}) \\
w \Vdash_g \mathtt{x} = \mathtt{y} + 1 \quad & \overset{\text{def}}{\Leftrightarrow} \quad g(\mathtt{x}) = g(\mathtt{y}) + 1 \\
w \Vdash_g f_i(\mathtt{x}+k_1) \simeq f_j(\mathtt{x}+k_2) & \overset{\text{def}}{\Leftrightarrow} \\
& \quad g(\mathtt{x}) + \min(k_1, k_2) \geqslant 0 \text{ and} \\
& \quad w(g(\mathtt{x})+k_1)(i) = w(g(\mathtt{x})+k_2)(j) \\
w \Vdash_g f_i(\mathtt{x}+k_1) \prec f_j(\mathtt{x}+k_2) & \overset{\text{def}}{\Leftrightarrow} \\
& \quad g(\mathtt{x}) + \min(k_1, k_2) \geqslant 0 \text{ and} \\
& \quad w(g(\mathtt{x})+k_1)(i) < w(g(\mathtt{x})+k_2)(j) \\
w \Vdash_g f_i(\mathtt{x}+k_1) \simeq d \quad & \overset{\text{def}}{\Leftrightarrow} \quad g(\mathtt{x}) + k_1 \geqslant 0 \text{ and} \\
& \quad w(g(\mathtt{x})+k_1)(i) = d \\
w \Vdash_g \exists \mathtt{x}\, \varphi \quad & \overset{\text{def}}{\Leftrightarrow} \quad \text{there is } k \geqslant 0 \text{ s.t. } w \Vdash_{g'} \varphi,
\end{aligned}$$

where g' is exactly as g except that $g'(\mathtt{x}) = k$. It is worth noting that the logic $\mathsf{FO}^{\mathrm{loc}}(\beta)$ contains two types of atomic formulae, those expressing constraints between positions (e.g. $\mathtt{x} = \mathtt{y} + 1$) and those expressing constraints between data values (e.g. $f_1(\mathtt{x}) \simeq f_3(\mathtt{x}+2)$).

4.2 Symbolic assignments and translation

Below, we smoothly generalise what is done in Sec. 3.2 and in Sec. 3.3. Since $\mathbb{D}$ is infinite, we cannot take advantage of the existence of Lemma 5 and therefore the translation map and its correctness proof are a bit more complicated. Anyway, we rely on such previous sections whenever possible since the general idea of the translation that internalises the semantics of PLTL-A($\mathbb{D}$) is very similar.

A **symbolic assignment** is defined as an expression of the form $(\mathtt{x}, e, \mathtt{y})$, where $\mathtt{x}$ is a variable in PLTL-A($\mathbb{D}$), $\mathtt{y}$ is an individual variable in $\mathsf{FO}^{\mathrm{loc}}(\beta)$, and e is either a datum $d \in \mathbb{D}$ or a term of the form $\mathsf{X}^n\mathtt{x}'$. The **shift** of e, written *shift*(e), is n if e is of the form $\mathsf{X}^n\mathtt{x}'$, otherwise it is 0. A **symbolic context** $\mathcal{S}$ is defined as a finite sequence of symbolic assignments (possibly empty). Given an environment g, we say that $\mathcal{S}$ **is compatible with** g iff for every symbolic assignment $(\mathtt{x}, e, \mathtt{y})$ occuring in $\mathcal{S}$, we have $g(\mathtt{y}) + \mathit{shift}(e) \geqslant 0$.

Lemma 6 can be easily generalised. The only substantial difference rests on the requirement that $\mathcal{S}$ is compatible with g. Assuming that $\mathcal{S}$ is a symbolic context of the form $(\mathtt{x}_{\alpha_1}, e_1, \mathtt{y}_1), \ldots, (\mathtt{x}_{\alpha_N}, e_N, \mathtt{y}_N)$ and g is an environment such that $\mathcal{S}$ is compatible with g, the model *update*$(\sigma, \mathcal{S}, g)$ is defined according to the clauses below.

- *update*$(\sigma, \varepsilon, g) \overset{\text{def}}{=} \sigma$,
- if $N > 0$, then *update*$(\sigma, \mathcal{S}, g) \overset{\text{def}}{=}$

$$\mathit{update}(\sigma, \mathcal{S}[1, N-1], g)[\mathtt{x}_{i_N} \mapsto_{g(\mathtt{y}_N)} V_N],$$

where $V_N = \mathit{update}(\sigma, \mathcal{S}[1, N-1], g)(g(\mathtt{y}_N) + n)(\mathtt{x}')$, if $e_N = \mathsf{X}^n\mathtt{x}'$, and $V_N = d$ if $e_N = d$. By compatibility, we have $g(\mathtt{y}_N) + n \geqslant 0$.

We define a translation map $\mathbf{t}$ from $\mathsf{PLTL\text{-}A}(\mathbb{D})$ to $\mathsf{FO}^{\mathrm{loc}}(\beta)$ generalising that from PLTL into $\mathsf{FO}(\omega)$ (Kamp 1968) and what is done in Sec. 3.3. The new cases are for atomic formulae and for assignments.

- Again, $\mathbf{t}(\langle \mathtt{x}{:=}\mathsf{X}^n\mathtt{x}'\rangle\,\psi, \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} \mathbf{t}(\psi, \mathtt{y}, \mathcal{S}\cdot(\mathtt{x}, \mathsf{X}^n\mathtt{x}', \mathtt{y}))$. Similarly, $\mathbf{t}(\langle \mathtt{x}{:=}d\rangle\,\psi, \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} \mathbf{t}(\psi, \mathtt{y}, \mathcal{S}\cdot(\mathtt{x}, d, \mathtt{y}))$.
- Let us focus on the translation of $\mathsf{X}^n\mathtt{x}_i = \mathsf{X}^m\mathtt{x}_j$. If there is no k such that $\alpha_k = i$ or $\alpha_k = j$, then

$$\mathbf{t}(\mathsf{X}^n\mathtt{x}_i = \mathsf{X}^m\mathtt{x}_j, \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} f_i(\mathtt{y}+n) \simeq f_j(\mathtt{y}+m).$$

Otherwise, the translation distinguishes three cases. Indeed, either the variables $\mathtt{x}_i$ and $\mathtt{x}_j$ may be present in the $\mathtt{x}_{\alpha_k}$'s but not at the "positions" $\mathtt{y}+n$ and $\mathtt{y}+m$, respectively; or the last update is for $\mathtt{x}_i$ at position $\mathtt{y}+n$ (resp. $\mathtt{x}_j$ at position $\mathtt{y}+m$). Then, define $\mathbf{t}(\mathsf{X}^n\mathtt{x}_i = \mathsf{X}^m\mathtt{x}_j, \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=}$

$$\begin{array}{c}
\Big(\bigwedge_{k \text{ s.t. } \alpha_k=i}(\mathtt{y}_k \neq \mathtt{y}+n) \wedge \bigwedge_{k \text{ s.t. } \alpha_k=j}(\mathtt{y}_k \neq \mathtt{y}+m) \wedge \\
f_i(\mathtt{y}+n) \simeq f_j(\mathtt{y}+m)\Big) \vee \\
\bigvee_{k \text{ s.t. } \alpha_k=i}\Big(\bigwedge_{k' \text{ s.t. } \alpha_{k'}=i \text{ and } k'>k}(\mathtt{y}_{k'} \neq \mathtt{y}+n) \wedge \\
\bigwedge_{k' \text{ s.t. } \alpha_{k'}=j \text{ and } k'>k}(\mathtt{y}_{k'} \neq \mathtt{y}+m)\Big) \wedge \\
(\mathtt{y}_k = \mathtt{y}+n) \wedge \mathbf{t}_1(e_k, \mathcal{S}[1,k-1]) \\
\vee \\
\bigvee_{k \text{ s.t. } \alpha_k=j}\Big(\bigwedge_{k' \text{ s.t. } \alpha_{k'}=i \text{ and } k'>k}(\mathtt{y}_{k'} \neq \mathtt{y}+n) \wedge \\
\bigwedge_{k' \text{ s.t. } \alpha_{k'}=j \text{ and } k'>k}(\mathtt{y}_{k'} \neq \mathtt{y}+m)\Big) \wedge \\
(\mathtt{y}_k = \mathtt{y}+m) \wedge \mathbf{t}_2(e_k, \mathcal{S}[1,k-1]),
\end{array}$$

where $\mathbf{t}_1(e_k, \mathcal{S}[1,k-1])$ and $\mathbf{t}_2(e_k, \mathcal{S}[1,k-1])$ are defined as follows. If $e_k = d$ then $\mathbf{t}_1(e_k, \mathcal{S}[1,k-1])$ is equal to $f_i(\mathtt{y}+n) \simeq d$ and $\mathbf{t}_2(e_k, \mathcal{S}[1,k-1])$ is equal to $f_j(\mathtt{y}+m) \simeq d$. Otherwise, if $e_k = \mathsf{X}^{n'}\mathtt{x}_\ell$, then

$$\mathbf{t}_1(e_k, \mathcal{S}[1,k-1]) \stackrel{\text{def}}{=} \mathbf{t}(\mathsf{X}^{n+n'}\mathtt{x}_\ell = \mathsf{X}^m\mathtt{x}_j, \mathtt{y}, \mathcal{S}[1,k-1]),$$
$$\mathbf{t}_2(e_k, \mathcal{S}[1,k-1]) \stackrel{\text{def}}{=} \mathbf{t}(\mathsf{X}^n\mathtt{x}_i = \mathsf{X}^{m+n'}\mathtt{x}_\ell, \mathtt{y}, \mathcal{S}[1,k-1]).$$

Above, first-order terms of the form $\mathtt{y}+\ell$ with $\ell \in \mathbb{Z}$ are used. By adding the prefix $\exists\mathtt{x}_0 \ldots \exists\mathtt{x}_\ell(\mathtt{x}_0 = \mathtt{x}) \wedge (\mathtt{x}_1 = \mathtt{x}_0 + 1) \wedge \ldots \wedge (\mathtt{x}_\ell = \mathtt{x}_{\ell-1} + 1)$ into the formula right after every quantification of $\mathtt{x}$, and replacing atomic formula using $\mathtt{x}+\ell$ by $\mathtt{x}_\ell$, we capture the intended effect in $\mathsf{FO}^{\mathrm{loc}}(\beta)$, assuming $\ell \geqslant 0$. Hence, we can assume $\mathsf{FO}^{\mathrm{loc}}(\beta)$ admits such terms without increasing its expressive power; negative shifts are handled in a similar way.

The translation is defined by cases just as in Lemma 6. Apart from the two cases in which no update affects $\mathtt{x}_i$ nor $\mathtt{x}_j$, the last part of the translation states that whenever k is the last position of interest (e.g., it is the position of $\mathcal{S}$ where the last update of $\mathtt{x}_i$ appears w.r.t. the relative position $\mathtt{y}+n$), a recursive call is performed. In such a call, the value of $\mathsf{X}^n\mathtt{x}_i$ is replaced as indicated by e_k (if $e_k = \mathsf{X}^{n'}\mathtt{x}_\ell$, then a shift of $n+n'$ from $\mathtt{y}$ is needed). The translation of $\mathsf{X}^n\mathtt{x}_i < \mathsf{X}^m\mathtt{x}_j$ is as above, except that occurrences of $\simeq$ must be replaced by $<$.

- The translation of $\mathsf{X}^n\mathtt{x}_i = d$ is a simplified version of the previous case. If there is no k such that $\alpha_k = i$, then $\mathbf{t}(\mathsf{X}^n\mathtt{x}_i = d, \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} f_i(\mathtt{y}+n) \simeq d$. Otherwise,

$$\begin{array}{c}
\mathbf{t}(\mathsf{X}^n\mathtt{x}_i = d, \mathtt{y}, \mathcal{S}) \stackrel{\text{def}}{=} \Big(\bigwedge_{k \text{ s.t. } \alpha_k=i}(\mathtt{y}_k \neq \mathtt{y}+n) \wedge f_i(\mathtt{y}+n) \simeq d\Big) \\
\vee \bigvee_{k \text{ s.t. } \alpha_k=i}\Big(\bigwedge_{k' \text{ s.t. } \alpha_{k'}=i \text{ and } k'>k}(\mathtt{y}_{k'} \neq \mathtt{y}+n)\Big) \wedge \\
(\mathtt{y}_k = \mathtt{y}+n) \wedge \mathbf{t}'(e_k, \mathcal{S}[1,k-1]),
\end{array}$$

with $\mathbf{t}'(e_k, \mathcal{S}[1,k-1])$ defined as follows. In case that $e_k = d'$, if $d' = d$, then $\mathbf{t}'(e_k, \mathcal{S}[1,k-1])$ is equal to $\top$, otherwise it is equal to $\bot$. If $e_k = \mathsf{X}^{n'}\mathtt{x}_\ell$, then

$$\mathbf{t}'(e_k, \mathcal{S}[1,k-1]) \stackrel{\text{def}}{=} \mathbf{t}(\mathsf{X}^{n+n'}\mathtt{x}_\ell = d, \mathtt{y}, \mathcal{S}[1,k-1]).$$

Again, it is also possible to use shortcuts, for instance to translate E-formulae with the clause $\mathbf{t}(\mathsf{E}\psi, \mathtt{y}, \mathcal{S}) = \exists\mathtt{y}'\ \mathbf{t}(\psi, \mathtt{y}', \mathcal{S})$. By way of example, the translation of

$$\mathsf{E}\langle \mathtt{x}_1{:=}\mathsf{X}\mathtt{x}_2\rangle\,\mathsf{E}\langle \mathtt{x}_3{:=}\mathsf{X}\mathtt{x}_1\rangle\,\mathsf{X}\mathtt{x}_1 = \mathsf{X}\mathsf{X}\mathtt{x}_3$$

is provided below (after simplifications):

$$\exists\mathtt{y}_1\exists\mathtt{y}_2\,\big((\mathtt{y}_2 \neq \mathtt{y}_2+2) \wedge (\mathtt{y}_1 \neq \mathtt{y}_2+1) \wedge f_1(\mathtt{y}_2+1) \simeq f_3(\mathtt{y}_2+2)\big)$$
$$\vee\big((\mathtt{y}_2 \neq \mathtt{y}_2+2) \wedge (\mathtt{y}_1 = \mathtt{y}_2+1) \wedge f_2(\mathtt{y}_2+2) \simeq f_3(\mathtt{y}_2+2)\big).$$

Thus, we can establish the following crucial result with a proof generalising the proof of Lemma 7.

Lemma 9. *Assuming that φ contains at most β variables, we have φ is $\mathsf{PLTL\text{-}A}(\mathbb{D})$ satisfiable iff $\exists\mathtt{y}\ (\neg\exists\mathtt{y}'\ \mathtt{y}' < \mathtt{y}) \wedge \mathbf{t}(\varphi, \mathtt{y}, \varepsilon)$ is $\mathsf{FO}^{\mathrm{loc}}(\beta)$ satisfiable.*

Thus, the satisfiability problem for $\mathsf{FO}^{\mathrm{loc}}(\beta)$ with $\beta \geqslant 1$ is Tower-hard. Indeed, by combining Lemma 1, Thm. 2 and Lemma 9, we get Tower-hardness of $\mathsf{FO}^{\mathrm{loc}}(1)$.

4.3 How to decide $\mathsf{FO}^{\mathrm{loc}}(\beta)$ satisfiability

Below, we characterise the satisfiability status of some $\mathsf{FO}^{\mathrm{loc}}(\beta)$ formula by taking advantage of decision procedures to solve the satisfiability problem for $\mathsf{PLTL}(\mathbb{D})$.

Given an $\mathsf{FO}^{\mathrm{loc}}(\beta)$ formula φ, typically built over atomic formulae of the form $f_i(\mathtt{x}+k_1) \simeq f_j(\mathtt{x}+k_2)$, $f_i(\mathtt{y}+k_1) < f_j(\mathtt{y}+k_2)$ and $f_i(\mathtt{z}+k_1) \simeq d$, we start by viewing it as an $\mathsf{FO}(\omega)$ formula. Indeed, such atomic formulae can be understood as $\mathsf{FO}(\omega)$ atomic formulae of the form $\mathtt{p}(\mathtt{x})$, $\mathtt{q}(\mathtt{y})$ and $\mathtt{r}(\mathtt{z})$. Here, the unary predicate $\mathtt{p}$ (resp. $\mathtt{q}$, $\mathtt{r}$) is parameterised by $\simeq, i, k_1, j, k_2$ (resp. by $<, i, k_1, j, k_2$ and d, i, k_1). By way of example, below we write $\mathtt{p}_{\simeq,i,k_1,j,k_2}$, $\mathtt{p}_{<,i,k_1,j,k_2}$ and $\mathtt{p}_{d,i,k_1}$ to denote such unary predicate symbols to relate atomic formulae in $\mathsf{FO}^{\mathrm{loc}}(\beta)$ and their reading in $\mathsf{FO}(\omega)$.

Kamp's Theorem (Kamp 1968; Rabinovich 2014) states that, given an $\mathsf{FO}(\omega)$ formula $\varphi(\mathtt{y})$ with free variable $\mathtt{y}$, one can effectively construct an PLTL formula ψ such that for all models $\sigma : \mathbb{N} \to (\mathsf{Prop} \to \{0,1\})$ and for all $i \in \mathbb{N}$, we have $\sigma, i \Vdash \psi$ iff $\sigma \Vdash_{[\mathtt{y}\mapsto i]} \varphi(\mathtt{y})$. Given an PLTL formula ψ built over propositional variables of the form $\mathtt{p}_{\simeq,i,k_1,j,k_2}$, $\mathtt{p}_{<,i,k_1,j,k_2}$ and $\mathtt{p}_{d,i,k_1}$, we write $\mathbf{t}_{cc}$ to denote the translation map from PLTL to $\mathsf{PLTL}(\mathbb{D})$ such that $\mathbf{t}_{cc}(\psi)$ is homomorphic for Boolean and temporal connectives, and

$$\begin{array}{lcl}
\mathbf{t}_{cc}(\mathtt{p}_{\simeq,i,k_1,j,k_2}) & \stackrel{\text{def}}{=} & \mathsf{X}^{k_1}\mathtt{x}_i = \mathsf{X}^{k_2}\mathtt{x}_j, \\
\mathbf{t}_{cc}(\mathtt{p}_{<,i,k_1,j,k_2}) & \stackrel{\text{def}}{=} & \mathsf{X}^{k_1}\mathtt{x}_i < \mathsf{X}^{k_2}\mathtt{x}_j, \\
\mathbf{t}_{cc}(\mathtt{p}_{d,i,k_1}) & \stackrel{\text{def}}{=} & \mathsf{X}^{k_1}\mathtt{x}_i = d.
\end{array}$$

We say that $\mathbf{t}_{cc}(\psi)$ is the **concretisation** of ψ.

Lemma 10. *Let $\varphi(\mathtt{y})$ be an $\mathsf{FO}^{loc}(\beta)$ formula with one free variable $\mathtt{y}$ and ψ be some PLTL formula equivalent to $\varphi(\mathtt{y})$ read in $\mathsf{FO}(\omega)$. Then, $\varphi(\mathtt{y})$ is $\mathsf{FO}^{loc}(\beta)$ satisfiable iff $\mathsf{E}\,\mathbf{t}_{cc}(\psi)$ is $\mathsf{PLTL}(\mathbb{D})$ satisfiable.*

Proof. We write Prop' to denote the finite set of propositional variables of one of the forms $\mathtt{p}_{\simeq,i,k_1,j,k_2}$, $\mathtt{p}_{<,i,k_1,j,k_2}$ and $\mathtt{p}_{d,i,k_1}$ obtained from the atomic formulae occurring in $\varphi(\mathtt{y})$. The following statements can be shown equivalent.

1. There are an $\mathsf{FO}^{\text{loc}}(\beta)$ model $w \in (\mathbb{D}^\beta)^\omega$ and an environment g such that $w \Vdash_g \varphi(\mathtt{y})$ in $\mathsf{FO}^{\text{loc}}(\beta)$.
2. There are an $\mathsf{FO}(\omega)$ model σ over the set Prop' of propositional variables, an environment g such that $\sigma \Vdash_g \varphi(\mathtt{y})$ in $\mathsf{FO}(\omega)$ and an $\mathsf{FO}^{\text{loc}}(\beta)$ model $w \in (\mathbb{D}^\beta)^\omega$ such that
   (a) if $\mathtt{p}_{\simeq,i,k_1,j,k_2} \in \mathsf{Prop}'$, then $\sigma(\alpha)(\mathtt{p}_{\simeq,i,k_1,j,k_2}) = 1$ iff $w(\alpha+k_1)(i) \simeq w(\alpha+k_2)(j)$, for all $\alpha \in \mathbb{N}$,
   (b) if $\mathtt{p}_{<,i,k_1,j,k_2} \in \mathsf{Prop}'$, then $\sigma(\alpha)(\mathtt{p}_{\simeq,i,k_1,j,k_2}) = 1$ iff $w(\alpha+k_1)(i) \prec w(\alpha+k_2)(j)$, for all $\alpha \in \mathbb{N}$,
   (c) if $\mathtt{p}_{d,i,k_1} \in \mathsf{Prop}'$, then $\sigma(\alpha)(\mathtt{p}_{\simeq,i,k_1,j,k_2}) = 1$ iff $w(\alpha+k_1)(i) = d$, for all $\alpha \in \mathbb{N}$.
3. There are an PLTL model σ over Prop' and a position γ such that $\sigma, \gamma \Vdash \psi$ and an $\mathsf{FO}^{\text{loc}}(\beta)$ model $w \in (\mathbb{D}^\beta)^\omega$ such that (a), (b) and (c) above. (by Kamp's Theorem)
4. There is an $\mathsf{PLTL}(\mathbb{D})$ model σ such that $\sigma, 0 \Vdash \mathsf{E}\,\mathbf{t}_{cc}(\psi)$. (by the semantics of $\mathsf{PLTL}(\mathbb{D})$) □

By Prop. 1, Lemma 10 and the effective construction of ψ from $\varphi(\mathtt{y})$ (see e.g. (Rabinovich 2014)), we can conclude.

Theorem 3. *Satisfiability problem for $\mathsf{FO}^{loc}(\beta)$ is decidable.*

If it is possible to build ψ in time $exp(Q(|\varphi(\mathtt{y})|), |\varphi(\mathtt{y})|)$, for some elementary function Q and $|\varphi(\mathtt{y})|$ denoting the size of $\varphi(\mathtt{y})$, then $\mathsf{FO}^{\text{loc}}(\beta)$ is in Tower (Schmitz 2016). Here *exp* is the function defined by induction on the first argument with $exp(0,n) = n$ and $exp(m+1,n) = 2^{exp(m,n)}$. Currently, Tower-membership is open.

Interestingly, $\mathsf{FO}^{\text{loc}}(\beta)$ can be viewed as a first-order logic over the concrete domain $\mathbb{D}$, following the terminology from (Baader and De Bortoli 2024). By way of example, the formula $f(\mathtt{x}+k) \simeq f'(\mathtt{x}+k')$ in $\mathsf{FO}^{\text{loc}}(\beta)$, corresponds to the atomic formula $= (succ^k \cdot f, succ^{k'} \cdot f')(\mathtt{x},\mathtt{x})$ in $\mathsf{FO}(\mathbb{D})$ from (Baader and De Bortoli 2024, Sec. 2), with the interpretation domain $\mathbb{N}$ (not understood as a concrete domain), $succ$ interpreted as the successor relation and the feature symbols f's having total interpretation. If the concrete domain $\mathbb{D}$ is either $\mathbb{N}$ or $\mathbb{Z}$, then $\mathbb{D}$ is not homomorphism ω-compact and therefore we cannot take advantage of (Baader and De Bortoli 2024, Thm. 3.1). Unfortunately, we cannot also take advantage of the decidability result in (Baader and De Bortoli 2024, Cor. 4.3) because $\mathbb{D}$ has a binary predicate.

4.4 $\mathsf{PLTL\text{-}A}(\mathbb{D})$ satisfiability is decidable

Now, we state the main corollaries of our previous results.

Corollary 1. *The satisfiability problem for $\mathsf{PLTL\text{-}A}(\mathbb{D})$ is decidable.*

From Lemmas 1 and 9 and Thm. 3 with $\beta = 1$, we get decidability of $\mathsf{PLTL\text{-}A}(\mathbb{D})$. Thus, adding modalities of the form $\langle \mathtt{x}{:=}\mathsf{X}^n\mathtt{y}\rangle$ to $\mathsf{PLTL}(\mathbb{D})$ preserves the decidability (see Prop. 1) but at the cost of moving from PSpace to Tower-hardness. The same argument applies if we consider finite traces instead of infinites ones. All the decidability/complexity results still hold true. We can also conclude by a characterisation of the expressive power of $\mathsf{PLTL\text{-}A}(\mathbb{D})$.

Corollary 2. *For every $\mathsf{PLTL\text{-}A}(\mathbb{D})$ formula φ, there is a $\mathsf{PLTL}(\mathbb{D})$ formula φ' logically equivalent to φ.*

Since $\mathsf{PLTL}(\mathbb{D})$ is a syntactic fragment of $\mathsf{PLTL\text{-}A}(\mathbb{D})$, we conclude that $\mathsf{PLTL\text{-}A}(\mathbb{D})$ and $\mathsf{PLTL}(\mathbb{D})$ have the same expressive power. The reasoning chain is summarised below.

$$\mathsf{PLTL\text{-}A}(\mathbb{D}) \xrightarrow{\text{Lemma 9}} \mathsf{FO}^{\text{loc}}(\beta) \xrightarrow{\text{Abstraction}} \mathsf{FO}(\omega)$$

$$\mathsf{FO}(\omega) \xrightarrow{\text{Kamp's Theorem}} \mathsf{PLTL} \xrightarrow{\text{Concretisation}} \mathsf{PLTL}(\mathbb{D}).$$

Cor. 2 can be extended to other concrete domains, as long as it is possible to define $\mathsf{FO}^{\text{loc}}(\beta)$ over symbolic models. This does not imply necessarily the decidability of $\mathsf{PLTL\text{-}A}(\mathbb{D})$, since the satisfiability problem for $\mathsf{PLTL}(\mathbb{D})$ may not be decidable. However, the same techniques apply for instance for $\mathsf{PLTL\text{-}A}(\mathbb{D})$, where $\mathbb{D}$ is the concrete domain RCC8 with space regions in $\mathbb{R}^2$ equipped with topological relations between spatial regions, see e.g. (Wolter and Zakharyaschev 2000; Balbiani and Condotta 2002).

5 Concluding Remarks

We introduced a version of LTL modulo theories with the new modality $\langle \mathtt{x}{:=}\mathtt{t}\rangle$ performing local assignments, which amounts to change locally the value of the variable $\mathtt{x}$. The concrete domains are of the form $(\mathbb{D}, =, <, (=_d)_{d\in\mathbb{D}})$, including $\mathbb{D}$ in $\{\mathbb{N}, \mathbb{Z}, \mathbb{Q}\}$ as well as any finite domain $\{0, \ldots, k-1\}$, for some $k \geqslant 2$. We established the decidability of the satisfiability problem by reduction into a new first-order logic on multi-attributed data words. Decidability of this instrumental logic is obtained by translation into $\mathsf{PLTL}(\mathbb{D})$ and evoking Kamp's Theorem. The addition of such local assignments allows us to express concisely many properties and the satisfiability problem is still decidable. We have also shown how to restrict ourselves to syntactic fragments, typically the one with a single variable. Using Kamp's Theorem (again), we show that $\mathsf{PLTL}(\mathbb{D})$ is as expressive as $\mathsf{PLTL\text{-}A}(\mathbb{D})$. Our method can be used with finite traces or with other concrete domains such as RCC8.

This work can be continued in many ways. First, the complexity of $\mathsf{PLTL\text{-}A}(\mathbb{D})$ restricted to one variable and to adjacent assignments is open. More interestingly, it would be possible to have a change operator stating that there is a way to change $k \geqslant 1$ values in the model such that a given formula φ holds afterwards. As far as we can judge, the techniques developed herein cannot handle this and the decidability status of such a variant is unclear. Last but not least, it would be interesting to resolve the satisfiability problem for $\mathsf{PLTL\text{-}A}(\mathbb{N})$ or fragments using SMT techniques, see e.g. (Barrett and Tinelli 2018). This is already challenging with $\mathsf{PLTL}(\mathbb{N})$ (without assignments) as the set of symbolic models of formulae in $\mathsf{PLTL}(\mathbb{N})$ is not always ω-regular.

Acknowledgments

We would like to thank the anonymous reviewers for their comments and suggestions that helped us to improve the quality of the document. R. Fervari is supported by Agencia I+D+i grant PICT 2021-00400, the EU H2020 research and innovation programme under the Marie Skłodowska-Curie grant agreements 101008233 (MISSION), the IRP SINFIN, SeCyT-UNC grant 33620230100178CB, and as part of France 2030 program ANR-11-IDEX-0003.

References

Alur, R., and Henzinger, T. 1994. A really temporal logic. *Journal of the ACM* 41(1):181–204.

Areces, C., and ten Cate, B. 2007. Hybrid logics. In *Handbook of Modal Logic*, volume 3 of *Studies in logic and practical reasoning*. North-Holland. 821–868.

Areces, C.; Fervari, R.; and Hoffmann, G. 2015. Relation-changing modal operators. *Logic Journal of the IGPL* 23(4):601–627.

Aucher, G.; Balbiani, P.; Fariñas del Cerro, L.; and Herzig, A. 2009. Global and local graph modifiers. *Electronic Notes in Theoretical Computer Science* 231:293–307.

Aucher, G.; van Benthem, J.; and Grossi, D. 2018. Modal logics of sabotage revisited. *Journal of Logic and Computation* 28(2):269–303.

Baader, F., and De Bortoli, F. 2024. The abstract expressive power of first-order and description logics with concrete domains. In *SAC'24*, 754–761. ACM.

Baader, F., and Hanschke, P. 1991. A scheme for integrating concrete domains into concept languages. In *IJCAI'91*, 452–457.

Baader, F., and Rydval, J. 2022. Using model theory to find decidable and tractable description logics with concrete domains. *Journal of Automated Reasoning* 66(3):357–407.

Balbiani, P., and Condotta, J. 2002. Computational complexity of propositional linear temporal logics based on qualitative spatial or temporal reasoning. In *FroCoS'02*, volume 2309 of *LNAI*, 162–173. Springer.

Bansal, K.; Brochenin, R.; and Lozes, E. 2009. Beyond shapes: Lists with ordered data. In *FoSSaCS'09*, volume 5504 of *LNCS*, 425–439. Springer.

Barrett, C., and Tinelli, C. 2018. Satisfiability Modulo Theories. In *Handbook of Model Checking*. Springer. 305–343.

Belardinelli, F.; Boureanu, I.; Malvone, V.; and Rajaona, F. 2023. Automatically verifying expressive epistemic properties of programs. In *AAAI'23*, 6245–6252. AAAI Press.

Belardinelli, F.; Boureanu, I.; Malvone, V.; and Rajaona, F. 2025. An SMT-based approach to the verification of knowledge-based programs. *Formal Aspects of Computing* 37(1):3:1–3:24.

Beutner, R., and Finkbeiner, B. 2025. AutoHyper: leveraging language inclusion checking for hyperproperty model-checking. *International Journal on Software Tools for Technology Transfer*. To appear in special issue for TACAS'23.

Bhaskar, A., and Praveen, M. 2024. Realizability problem for constraint LTL. *I&C* 296:105126.

Bojańczyk, M.; David, C.; Muscholl, A.; Schwentick, T.; and Segoufin, L. 2011. Two-variable logic on data words. *ACM ToCL* 12(4):27.

Bollig, B.; Sangnier, A.; and Stietel, O. 2024. On the satisfiability of local first-order logics with data. *Logical Methods in Computer Science* 20(3).

Borgwardt, S.; Bortoli, F. D.; and Koopmann, P. 2024. The precise complexity of reasoning in $\mathcal{ALC}$ with ω-admissible concrete domains. In *DL'24*, volume 3739 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Büchi, J. 1962. On a decision method in restricted second-order arithmetic. In *International Congress on Logic, Method and Philosophical Science'60*, 1–11.

Carapelle, C. 2015. *On the satisfiability of temporal logics with concrete domains*. Ph.D. Dissertation, Leipzig University.

Catta, D.; Leneutre, J.; Malvone, V.; and Murano, A. 2024. Obstruction alternating-time temporal logic: A strategic logic to reason about dynamic models. In *AAMAS'24*, 271–280. International Foundation for Autonomous Agents and Multiagent Systems / ACM.

Cimatti, A.; Griggio, A.; Mover, S.; Roveri, M.; and Tonetta, S. 2022. Verification modulo theories. *Formal Methods in System Design* 60(3):452–481.

Czerwinski, W., and Orlikowski, L. 2021. Reachability in vector addition systems is Ackermann-complete. In *STOC'21*, 1229–1240. IEEE.

Decker, N.; Habermehl, P.; Leucker, M.; and Thoma, D. 2014. Ordered navigation on multi-attributed data words. In *CONCUR'14*, volume 8704 of *LNCS*, 497–511.

Demri, S., and Gascon, R. 2008. Verification of qualitative $\mathbb{Z}$ constraints. *TCS* 409(1):24–40.

Demri, S.; Lazić, R.; and Nowak, D. 2007. On the freeze quantifier in constraint LTL: decidability and complexity. *I&C* 205(1):2–24.

Diekert, V., and Gastin, P. 2008. First-order definable languages. In Flum, J.; Grädel, E.; and Wilke, T., eds., *Logic and Automata: History and Perspectives*, volume 2 of *Texts in Logic and Games*. Amsterdam University Press. 261–306.

Faella, M., and Parlato, G. 2024. A unified automata-theoretic approach to LTL_f modulo theories. In *ECAI'24*, 1254–1261.

Felli, P.; Montali, M.; and Winkler, S. 2022. Linear-time verification of data-aware dynamic systems with arithmetic. In *AAAI'22*, 5642–5650. AAAI Press.

Figueira, D., and Segoufin, L. 2009. Future-looking logics on data words and trees. In *MFCS'09*, volume 5734 of *LNCS*, 331–343. Springer.

Figueira, D. 2010. *Reasoning on words and trees with data*. Ph.D. Dissertation, ENS Cachan.

Fitting, M. 2002. Modal logic between propositional and first-order. *Journal of Logic and Computation* 12(6):1017–1026.

Gabbay, D.; Pnueli, A.; Shelah, S.; and Stavi, J. 1980. On the temporal analysis of fairness. In *POPL'80*, 163–173. ACM Press.

Galimullin, R.; Galdyshev, M.; Mittelmann, M.; and Motamed, N. 2025. Changing the rules of the game: reasoning about dynamic phenomena in multi-agent systems. In *AAMAS'25*, 829–838.

Geatti, L.; Gianola, A.; Gigante, N.; and Winkler, S. 2023. Decidable fragments of $\mathrm{LTL_f}$ modulo theories. In *ECAI'23*, volume 372, 811–818. IOS Press.

Geatti, L.; Gianola, A.; and Gigante, N. 2025. First-order automata. In *AAAI'25*, 14940–14948.

Gianola, A.; Montali, M.; and Winkler, S. 2025. SMT Techniques for Data-Aware Process Mining. *Künstliche Intelligenz*. To appear.

Halpern, J. 1995. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. *AI* 75(2):361–372.

Herzig, A.; Maris, F.; and Vianey, J. 2019. Dynamic logic of parallel propositional assignments and its applications to planning. In *IJCAI'19*, 5576–5582. ijcai.org.

Herzig, A. 2014. Belief change operations: A short history of nearly everything, told in dynamic logic of propositional assignments. In *KR'14*. AAAI Press.

Jurdziński, M., and Lazić, R. 2011. Alternating automata on data trees and XPath satisfiability. *ACM ToCL* 12(3):19:1–19:21.

Kamp, J. 1968. *Tense Logic and the theory of linear order*. Ph.D. Dissertation, UCLA, USA.

Labai, N.; Ortiz, M.; and Simkus, M. 2020. An Exptime Upper Bound for $\mathcal{ALC}$ with integers. In *KR'20*, 425–436. Morgan Kaufman.

Leroux, J., and Schmitz, S. 2019. Reachability in vector addition systems is primitive-recursive in fixed dimension. In *LiCS'19*, 1–13. IEEE.

Leroux, J. 2021. The reachability problem for Petri nets is not primitive recursive. In *STOC'21*, 1241–1252. IEEE.

Lutz, C. 2002a. Adding numbers to the SHIQ description logic: First results. In *KR'02*, 191–202. Morgan Kaufmann.

Lutz, C. 2002b. *The Complexity of Description Logics with Concrete Domains*. Ph.D. Dissertation, RWTH, Aachen.

Lutz, C. 2006. Complexity and succinctness of public announcement logic. In *AAMAS'06*, 137–143. ACM.

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

Rabinovich, A. 2014. A Proof of Kamp's theorem. *Logical Methods in Computer Science* 10(1):1–16.

Rodríguez, A., and Sánchez, C. 2024. Realizability modulo theories. *Journal of Logical and Algebraic Methods in Programming* 140:100971.

Schmitz, S. 2016. Complexity hierarchies beyond elementary. *TOCT* 8(1):3:1–3:36.

Segoufin, L., and Toruńczyk, S. 2011. Automata based verification over linearly ordered data domains. In *STACS'11*, 81–92.

Sistla, A., and Clarke, E. 1985. The complexity of propositional linear temporal logic. *Journal of the ACM* 32(3):733–749.

Spaan, E. 1993. *Complexity of Modal Logics*. Ph.D. Dissertation, ILLC, Amsterdam University.

Stockmeyer, L. J. 1974. *The complexity of decision problems in automata theory and logic*. Ph.D. Dissertation, Massachusetts Institute of Technology, USA.

van Ditmarsch, H.; Herzig, A.; and de Lima, T. 2012. Public announcements, public assignments and the complexity of their logic. *Journal of Applied Non-Classical Logics* 22(3):249–273.

Wolper, P. 1983. Temporal logic can be more expressive. *I&C* 56:72–99.

Wolter, F., and Zakharyaschev, M. 2000. Spatio-temporal representation and reasoning based on RCC-8. In *KR'00*, 3–14.