



HAL
open science

Computational Complexity of Standpoint LTL

Stéphane Demri, Przemyslaw Andrzej Walega

► **To cite this version:**

Stéphane Demri, Przemyslaw Andrzej Walega. Computational Complexity of Standpoint LTL. 27th European Conference on Artificial Intelligence, Oct 2024, Santiago de Compostela, Spain. pp.1206–1213. hal-04742082

HAL Id: hal-04742082

<https://hal.science/hal-04742082v1>

Submitted on 17 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Computational Complexity of Standpoint LTL

Stéphane Demri^a and Przemysław Andrzej Wałęga^b

^aUniversité Paris-Saclay, ENS Paris-Saclay, CNRS, LMF, 91190, Gif-sur-Yvette, France

^bUniversity of Oxford, Queen Mary University of London, United Kingdom

Abstract. Standpoint linear temporal logic SLTL is a recent formalism able to model possibly conflicting commitments made by distinct agents, taking into account aspects of temporal reasoning. In this paper, we analyse the computational properties of SLTL. First, we establish logarithmic-space reductions between the satisfiability problems for the multi-dimensional modal logic PTLxS5 and SLTL. This leads to the ExpSpace-completeness of the satisfiability problem in SLTL, which is a surprising result in view of previous investigations. Next, we present a method of restricting SLTL so that the obtained fragment is a strict extension of both the (non-temporal) standpoint logic and linear-time temporal logic LTL, but the satisfiability problem is PSpace-complete in this fragment. Thus, we show how to combine standpoint logic with LTL so that the worst-case complexity of the obtained combination is not higher than of pure LTL.

1 Introduction

Standpoint Logic. Recently, a new framework based on modal logics was developed in order to interpret languages in the presence of vagueness [21]. The framework is called ‘standpoint logic’ where standpoints (a concept first introduced by Bennett [10] in a logical context, see also [11]) are used to interpret vague expressions. Logical reasoning about vagueness has a long tradition stemming from fuzzy logics [34, 35], to information logics based on rough sets [28, 27, 7]. In standpoint logic, each standpoint s is associated with modalities \diamond_s and \square_s , and with a set of interpretations (a.k.a. *precisifications*) corresponding to s . While $\diamond_s\varphi$ reads as “according to s , it is conceivable that φ ”, dually, $\square_s\varphi$ reads as “according to s , it is unequivocal that φ ”. The language of standpoint logic is equipped also with a binary operator \preceq between standpoints, such that $s \preceq s'$ is interpreted as “the standpoint s is sharper than s' ” leading to the validity of the modal axiom $\square_{s'}p \rightarrow \square_s p$. This is reminiscent to partially-ordered (S4) modal operators [1], role hierarchies in description logics [4], and more generally to grammar logics [17], for which computational properties are well studied [6, 14]. Originally, standpoint logic framework was developed for propositional calculus [21], then also for predicate logic and description logics [23, 24], and most recently for the temporal logic LTL [20]. This framework has a potential for further combinations with logical formalisms dedicated to knowledge representation and reasoning, allowing us to perform logical reasoning about vagueness [18, 9, 11].

Standpoint LTL. As evoked above, the recent paper [20] introduced a multi-perspective approach by combining standpoints and temporal reasoning expressed in the linear-time temporal logic LTL [29], which is a very popular specification language, for instance used for model-checking [5], temporal planning [13, 2], and

temporal reasoning with description logics [3]. This new formalism, called SLTL, handles both evolutions of systems and changes of standpoints. Gigante et al. [20] develop a tableau-based proof system to reason about SLTL, leading to a computational analysis for deciding the satisfiability status of SLTL formulae. Within SLTL, each standpoint is interpreted as a set of LTL models (i.e. as a set of ω -sequences of propositional valuations, also known as traces) and the logical formalism has the ability to model possibly conflicting commitments made by distinct agents. Hence, SLTL significantly increases the modelling capabilities offered by LTL, and so, SLTL can be seen as a non-trivial extension of LTL. Our initial motivation in this work is to understand the computational properties of SLTL. We agree with Gigante et al. [20] that it is particularly desirable to be able to decide the satisfiability status of SLTL formulae in polynomial-space, which is the best we can hope for in view of PSpace-completeness of LTL [31]. However, in contrast to the results of Gigante et al. [20], we show that the PSpace membership can be guaranteed only for strict fragments of SLTL (see Section 4) and ExpSpace is required for the full SLTL (see Section 3).

Our contributions. We study the computational properties of the satisfiability problem for the standpoint linear temporal logic SLTL. We show that the problem is ExpSpace-complete (Theorem 5) by establishing logarithmic-space reductions between SLTL and the multi-dimensional modal logic PTLxS5 whose satisfiability problem is known to be ExpSpace-complete [19]. The obtained ExpSpace-completeness of SLTL contrasts with PSpace-membership claimed by Gigante et al. [20]. In Section 4.1, we provide examples of SLTL formulae whose satisfiability is challenging to check as their models have an infinite set of traces and at every position, an exponential amount of valuations are witnessed on such traces (see Proposition 6). In Section 4.2, we identify a fragment of SLTL which contains both LTL and propositional standpoint logic, but the satisfiability problem can be decided in polynomial-space. Since LTL is known to be PSpace-complete [31], we obtain the same tight complexity result for the newly introduced fragment of SLTL (Theorem 11). To do so, we use the automata-based approach following the general principles for LTL [32], but with non-trivial modifications.

2 Logical Preliminaries

In this section, we briefly introduce the standpoint linear temporal logic SLTL, the propositional standpoint logic PSL, as well as the multi-dimensional modal logic PTLxS5, which we exploit in Section 3. For motivations and detailed presentation of these logics, we refer a reader to [22, 20] and [19, Chapter 5].

2.1 Standpoint linear temporal logic SLTL

The SLTL formulae are built over a countably infinite set \mathcal{P} of *propositional variables* and a countably infinite set $\mathcal{S} = \{\mathbf{s}, \mathbf{s}', \dots\}$ of *standpoint symbols* including the *universal standpoint symbol* $*$. The SLTL formulae φ are defined according to the grammar:

$$\varphi ::= p \mid \mathbf{s} \preceq \mathbf{s}' \mid \neg\varphi \mid \varphi \wedge \varphi \mid \diamond_s \varphi \mid \square_s \varphi \mid X\varphi \mid \varphi U \varphi,$$

where $p \in \mathcal{P}$ and $\mathbf{s}, \mathbf{s}' \in \mathcal{S}$; other Boolean connectives and LTL temporal operators (e.g. $\rightarrow, \leftrightarrow, \vee, G$ for “always in the future”, and F for “sometime in the future”) are treated as standard abbreviations. In terms of expressivity, one modality among \diamond_s, \square_s is sufficient. An SLTL model, \mathcal{M} , is a structure of the form $\mathcal{M} = (\Pi, \lambda)$ where,

- $\Pi \neq \emptyset$ is a set of LTL models (traces) of the form $\sigma : \mathbb{N} \rightarrow 2^{\mathcal{P}}$,
- λ is a map of the form $\lambda : \mathcal{S} \rightarrow (2^{\Pi} \setminus \{\emptyset\})$ such that $\lambda(*) = \Pi$.

For example, Figure 1 presents an SLTL model with six traces. The *satisfaction relation*, $\mathcal{M}, \sigma, i \models \varphi$, for an SLTL model $\mathcal{M} = (\Pi, \lambda)$, $\sigma \in \Pi$, $i \in \mathbb{N}$, and an SLTL formula φ is defined inductively as follows (we omit the standard clauses for Boolean connectives):

$$\begin{aligned} \mathcal{M}, \sigma, i \models p & \stackrel{\text{def}}{\iff} p \in \sigma(i), \\ \mathcal{M}, \sigma, i \models \mathbf{s} \preceq \mathbf{s}' & \stackrel{\text{def}}{\iff} \lambda(\mathbf{s}) \subseteq \lambda(\mathbf{s}'), \\ \mathcal{M}, \sigma, i \models \diamond_s \varphi & \stackrel{\text{def}}{\iff} \mathcal{M}, \sigma', i \models \varphi, \text{ for some } \sigma' \in \lambda(\mathbf{s}), \\ \mathcal{M}, \sigma, i \models \square_s \varphi & \stackrel{\text{def}}{\iff} \mathcal{M}, \sigma', i \models \varphi, \text{ for all } \sigma' \in \lambda(\mathbf{s}), \\ \mathcal{M}, \sigma, i \models X\varphi & \stackrel{\text{def}}{\iff} \mathcal{M}, \sigma, i+1 \models \varphi, \\ \mathcal{M}, \sigma, i \models \varphi U \psi & \stackrel{\text{def}}{\iff} \text{there is } i' \geq i \text{ such that } \mathcal{M}, \sigma, i' \models \psi \\ & \text{and } \mathcal{M}, \sigma, i'' \models \varphi \text{ for all } i \leq i'' < i'. \end{aligned}$$

The satisfiability problem for SLTL takes as input an SLTL formula φ and asks whether there is a model $\mathcal{M} = (\Pi, \lambda)$ and $\sigma \in \Pi$ such that $\mathcal{M}, \sigma, 0 \models \varphi$. By way of example, we provide an SLTL formula below taken from the medical devices example [20, Section 2.1] (more elaborated examples can be found in [21, 22, 20]):

$$\square_*(G \neg \text{Malf} \rightarrow \text{Test}) \wedge \square_{IT}(\text{Comp} \vee \text{Test} \rightarrow \text{Safe}).$$

The first conjunct states that all countries agree in their standpoints that if a medical device never malfunctions, then it is safe according to testing. The second one states that Italy deems a device safe if it is safe according to testing or it has been found safe by comparison.

Regarding our definition of SLTL, it is worth observing that, as far as we can judge, the satisfiability problem is not formally defined in [20, Section 2.2]. Only the validity problem is defined. In particular, non validity of the SLTL formula φ is defined as the existence of an SLTL model $\mathcal{M} = (\Pi, \lambda)$ and $\sigma \in \Pi$ such that $\mathcal{M}, \sigma, 0 \models \neg\varphi$. So, our definition of satisfiability is dual to the notion of validity used by Gigante et al. [20].

Besides, it is worth noting that the above presentation of SLTL differs slightly with the definition of Gigante et al. [20, Section 2.2], but it has no impact on our results, as described next. SLTL formulae, as defined in [20], are in negation normal form and allow for using the “release” LTL operator. In our definition negation is unrestricted and we do not use the “release” operator, which makes no substantial difference. Gigante et al. [20] assumes also that the formulae $\mathbf{s} \preceq \mathbf{s}'$ cannot be combined with other formulae, so the way we define SLTL formulae is slightly more expressive. However, our ExpSpace-hardness proof (reduction in Lemma 1) does not use formulae of the form $\mathbf{s} \preceq \mathbf{s}'$ (actually only the modalities \square_*, X , and

U are needed). The ExpSpace-membership (Corollary 4) for our, slightly richer language, clearly implies the same upper bound for the weaker language of Gigante et al. [20].

The change of standpoint performed with the modalities \diamond_s and \square_s is reminiscent to the change of observational power studied in [8] with the modalities Δ^o . In both cases, a modality explicitly performs a change in the way the forthcoming formulae are evaluated.

2.2 Propositional standpoint logic PSL

In the sequel, we also consider *propositional standpoint logic* [22] (herein, written PSL) understood as the fragment of SLTL without temporal connectives. The grammar of formulae is restricted to

$$\varphi ::= p \mid \mathbf{s} \preceq \mathbf{s}' \mid \neg\varphi \mid \varphi \wedge \varphi \mid \diamond_s \varphi \mid \square_s \varphi,$$

and the models are of the form $\mathcal{M} = (\Pi, V)$ where Π is a finite non-empty set of *precisifications*, $V : \mathcal{S} \cup \mathcal{P} \rightarrow 2^{\Pi}$ is a *valuation* such that for all $\mathbf{s} \in \mathcal{S}$, we have $V(\mathbf{s}) \neq \emptyset$ and $V(*) = \Pi$. The satisfaction relation is defined as follows (where $\pi \in \Pi$ and we omit the obvious clauses for Boolean connectives):

$$\begin{aligned} \mathcal{M}, \pi \models p & \stackrel{\text{def}}{\iff} \pi \in V(p), \\ \mathcal{M}, \pi \models \mathbf{s} \preceq \mathbf{s}' & \stackrel{\text{def}}{\iff} V(\mathbf{s}) \subseteq V(\mathbf{s}'), \\ \mathcal{M}, \pi \models \diamond_s \varphi & \stackrel{\text{def}}{\iff} \mathcal{M}, \pi' \models \varphi, \text{ for some } \pi' \in V(\mathbf{s}), \\ \mathcal{M}, \pi \models \square_s \varphi & \stackrel{\text{def}}{\iff} \mathcal{M}, \pi' \models \varphi, \text{ for all } \pi' \in V(\mathbf{s}). \end{aligned}$$

The satisfiability problem, for an input formula φ , consists in checking whether there is some pair \mathcal{M}, π such that $\mathcal{M}, \pi \models \varphi$. This problem is NP-complete. To show NP-membership, Gómez Álvarez [21, Section 4.4.2] proved that if a satisfiable formula φ contains N_1 many standpoint symbols and N_2 many diamond modal operators, then φ is satisfied in a model $\mathcal{M} = (\Pi, V)$ such that $|\Pi| \leq N_1 + N_2 + 1$. An alternative way to get the NP-membership is to translate φ into a formula $\bigwedge_s \diamond_s \mathbf{t}(\varphi)$ of the modal logic S5 [12], where \mathbf{t} turns standpoint operators into modal operators in the following manner: $\mathbf{t}(\mathbf{s} \preceq \mathbf{s}') = \square(\mathbf{s} \rightarrow \mathbf{s}')$, $\mathbf{t}(\diamond_s \varphi) = \diamond(\mathbf{s} \wedge \mathbf{t}(\varphi))$, and $\mathbf{t}(\square_s \varphi) = \square(\mathbf{s} \rightarrow \mathbf{t}(\varphi))$. The correctness of such a reduction relies naturally on the Kripke-style semantics for PSL. We will refine complexity analysis of PSL in Section 4.2, which will be essential to establish complexity of SLTL fragments.

2.3 Multi-dimensional modal logic PTLxS5

Another logic that is useful herein is the multi-dimensional modal logic PTLxS5 [19, Chapter 5] defined as the product of LTL and S5. PTLxS5 formulae are generated from the grammar

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \diamond\varphi \mid \square\varphi \mid X\varphi \mid \varphi U \varphi,$$

where $p \in \mathcal{P}$ is a propositional variable. As in SLTL, we use standard abbreviations for other Boolean connectives and LTL operators (\rightarrow, \vee, G, F , etc.). The *models* for PTLxS5 are of the form $\mathcal{M} = (\mathbb{N} \times W, R, L)$ where (W, R) is an S5-frame (i.e. R is an equivalence relation on W) and $L : \mathbb{N} \times W \rightarrow 2^{\mathcal{P}}$. The satisfaction relation for PTLxS5 is defined as follows (again, we omit the

standard clauses for Boolean connectives):

$$\begin{aligned}
\mathcal{M}, (n, w) \models p &\stackrel{\text{def}}{\iff} p \in L(n, w), \\
\mathcal{M}, (n, w) \models \diamond\varphi &\stackrel{\text{def}}{\iff} \mathcal{M}, (n, w') \models \varphi, \text{ for some } w' \in R(w), \\
\mathcal{M}, (n, w) \models \square\varphi &\stackrel{\text{def}}{\iff} \mathcal{M}, (n, w') \models \varphi, \text{ for all } w' \in R(w), \\
\mathcal{M}, (n, w) \models X\varphi &\stackrel{\text{def}}{\iff} \mathcal{M}, (n+1, w) \models \varphi, \\
\mathcal{M}, (n, w) \models \varphi U\psi &\stackrel{\text{def}}{\iff} \text{there is } n' \geq n \text{ such that } \mathcal{M}, (n', w) \models \psi \\
&\quad \text{and } \mathcal{M}, (n'', w) \models \varphi \text{ for all } n \leq n'' < n'.
\end{aligned}$$

Therefore, the modalities \diamond and \square allow us to move within the (W, R) dimension whereas the temporal connectives X and U allow us to move along the (\mathbb{N}, \leq) dimension. The satisfiability problem for PTLxS5 takes as input a PTLxS5 formula φ and asks whether there is a model $\mathcal{M} = (\mathbb{N} \times W, R, L)$ and $(n, w) \in \mathbb{N} \times W$ such that $\mathcal{M}, (n, w) \models \varphi$. It is known, that satisfaction of a formula can be always witnessed by a model $\mathcal{M} = (\mathbb{N} \times W, R, L)$ with $R = W \times W$ and by a pair $(0, w)$ (i.e. its first component is the origin position 0). In the sequel we will use this assumption; in particular, we will assume that $R = W \times W$, and for simplicity of presentation we will drop the component R from PTLxS5 models.

It is worth noting that the above presentation of PTLxS5 differs slightly from the definitions by Gabbay et al. [19, Section 2.1], but it has no impact on our results. Indeed, Gabbay et al. [19] use only the strict “until” operator, which we denote by $U_{<}$ (and no next-time operator X) and whose semantics is as follows:

$$\begin{aligned}
\mathcal{M}, (n, w) \models \varphi U_{<} \psi &\stackrel{\text{def}}{\iff} \text{there is } n' > n \text{ with } \mathcal{M}, (n', w) \models \psi \\
&\quad \text{and } \mathcal{M}, (n'', w) \models \varphi \text{ for all } n < n'' < n'.
\end{aligned}$$

It is easy to see that $\varphi U_{<} \psi$ can be encoded by $X(\varphi U\psi)$ and therefore the ExpSpace-hardness for PTLxS5 proved by Gabbay et al. [19, Theorem 5.43] applies also to our version of PTLxS5. As far as the ExpSpace-membership is concerned, the satisfiability problem for our version of PTLxS5 is in ExpSpace using the approach of Gabbay et al. [19, Theorem 6.65] dedicated to PTLxS5 with strict “until” and using a standard renaming technique [19, Proposition 2.10]. Note that a naive translation \mathfrak{t} from our language to a formula with strict “until” exploiting $\mathfrak{t}(\varphi U\psi) = \mathfrak{t}(\psi) \vee (\mathfrak{t}(\varphi) \wedge \mathfrak{t}(\varphi) U_{<} \mathfrak{t}(\psi))$, would cause an exponential blow-up. However, we can get a logarithmic-space reduction using the renaming technique, where any subformula χ is associated with a fresh propositional variable p_χ . For instance, to capture the meaning of $\varphi U\psi$ we introduce an additional formula $\square G(p_{\varphi U\psi} \leftrightarrow p_\psi \vee (p_\varphi \wedge p_\varphi U_{<} p_\psi))$. This additional formula (propagating an equivalence all over the model), if asserted in any world of the form $(0, w)$, allows us to state that $p_{\varphi U\psi}$ is equivalent with $\psi \vee (\varphi \wedge \varphi U_{<} \psi)$, in all elements in $\mathbb{N} \times W$. The propagation is over the model because of the modality $\square G$ and we can always assume that a world satisfying our formula is of the form $(0, w)$. As a conclusion, the version of PTLxS5 involved in this paper admits also an ExpSpace-complete satisfiability problem (ExpSpace-hardness follows from the fact that $\varphi U_{<} \psi$ can be encoded by $X(\varphi U\psi)$).

Let us conclude this section by evoking the relationships between SLTL and the well-known modal logic S5 [12]. The logic SLTL contains a modality \square_* where $*$ can be understood as the universal standpoint interpreted by the total set of traces and therefore \square_* behaves naturally as an S5 modality, whence the component S5 in PTLxS5. The presence of S5 is not our finding as it has been already observed that quantification over precisifications leads to S5 modalities, see e.g., the works of Bennett [9, Section 2.1] and Bennett [10, page 43],

as well as the presence of S5 modalities for modelling standpoints in [21, Section 3.5] and in [21, Chapter 4]. Furthermore, the relationship with multi-dimensional modal logics is already briefly evoked in [21, Section 7.3.3]. More importantly, an early introduction of some multi-dimensional modal logic is performed by Bennett [9, Section 2.1] where first-order logic and propositional modal logic S5 are mixed. Hence, the fact that we use PTLxS5 is not a total surprise, and the need for multi-dimensional modal logics was in the air for some time. Our contribution consists in establishing a formal result involving multi-dimensional modal logics and in designing simple logarithmic-space reductions between SLTL and PTLxS5, proving ExpSpace-hardness of SLTL, despite the complexity result claimed in [20, Theorem 28]. The design of a PSpace fragment completes our analysis and provides us with a fragment of SLTL which is a strict extension of both LTL and standpoint logic, but its computational complexity is not higher than the complexity of pure LTL.

3 Satisfiability for SLTL is ExpSpace-complete

In this section we design logarithmic-space translations from SLTL to PTLxS5 formulae, and vice versa. As a result, we will obtain that the computational complexity of satisfiability for SLTL formulae is the same as for PTLxS5 formulae, namely ExpSpace-complete. Both of our translations exploit similarities between SLTL and PTLxS5 models. More specifically, we will consider an element $w \in W$ in an PTLxS5 model $(\mathbb{N} \times W, L)$ as a name for an LTL trace $\sigma \in \Pi$ from an SLTL model (Π, λ) .

3.1 Translation from PTLxS5 to SLTL

Let φ be a PTLxS5 formula and $\mathfrak{t}_1(\varphi)$ be its translation obtained from φ by replacing every occurrence of \diamond by \diamond_* and every occurrence \square by \square_* (an alternative translation consists in using \diamond_s and \square_s for some fixed $s \in S$). We show that this simple translation preserves satisfiability.

Lemma 1. φ is PTLxS5-satisfiable iff $\mathfrak{t}_1(\varphi)$ is SLTL-satisfiable.

Proof sketch. Assume that there is a PTLxS5 model $\mathcal{M} = (\mathbb{N} \times W, L)$ and $(0, w) \in \mathbb{N} \times W$ such that $\mathcal{M}, (0, w) \models \varphi$. Let us build an SLTL model $\mathcal{M}' = (\Pi, \lambda)$ and a trace $\sigma \in \Pi$ such that $\mathcal{M}, \sigma, 0 \models \mathfrak{t}_1(\varphi)$. To this end, we let Π consist of all traces of the form

$$\sigma_{w'} \stackrel{\text{def}}{=} L((0, w'), L((1, w')), L((2, w')), L((3, w')), \dots$$

with $w' \in W$; note that we represent a trace $\sigma' : \mathbb{N} \rightarrow 2^P$ as an ω -sequence $\sigma'(0), \sigma'(1), \sigma'(2), \dots$. We let $\lambda(s) \stackrel{\text{def}}{=} \Pi$, for each $s \in S$ (for the construction it is only important that $\lambda(*) = \Pi$ as no other standpoint symbol occurs in $\mathfrak{t}_1(\varphi)$). We can prove by structural induction that, for all subformulae ψ of φ and for all $(n, w') \in \mathbb{N} \times W$, we have $\mathcal{M}, (n, w') \models \psi$ iff $\mathcal{M}', \sigma_{w'}, n \models \mathfrak{t}_1(\psi)$. By way of example we handle below the case when the subformula is of the form $\square\psi$. It suffices to observe that the following statements are equivalent:

- $\mathcal{M}, (n, w') \models \square\psi$
- $\mathcal{M}, (n, w'') \models \psi$ for all $w'' \in W$ (by definition of \models)
- $\mathcal{M}', \sigma_{w''}, n \models \mathfrak{t}_1(\psi)$ for all $w'' \in W$ (by induction hypothesis)
- $\mathcal{M}', \sigma', n \models \mathfrak{t}_1(\psi)$ for all $\sigma' \in \Pi$ (by definition of Π)
- $\mathcal{M}', \sigma_{w'}, n \models \square_* \mathfrak{t}_1(\psi)$ (by definition of \models)
- $\mathcal{M}', \sigma_{w'}, n \models \mathfrak{t}_1(\square\psi)$ (by definition of \mathfrak{t}_1)

Therefore, we can show that $\mathcal{M}, \sigma_w, 0 \models t_1(\varphi)$.

For the opposite implication, assume that there are an SLTL model $\mathcal{M} = (\Pi, \lambda)$ and a trace $\sigma \in \Pi$ such that $\mathcal{M}, \sigma, 0 \models t_1(\varphi)$. Let us build a PTLxS5 model $\mathcal{M}' = (\mathbb{N} \times W, L)$ and $(0, w) \in \mathbb{N} \times W$ such that $\mathcal{M}', (0, w) \models \varphi$. We let $W \stackrel{\text{def}}{=} \Pi$ and $L((n, \sigma')) \stackrel{\text{def}}{=} \sigma'(n)$, for all $(n, \sigma') \in \mathbb{N} \times W$. We can show by structural induction that for all subformulae ψ of φ , all $\sigma' \in \Pi$, and all $n \in \mathbb{N}$, we have $\mathcal{M}, \sigma', n \models t_1(\psi)$ iff $\mathcal{M}', (n, \sigma') \models \psi$. By way of example, we handle below the case when the subformula is of the form $\psi_1 \mathbf{U} \psi_2$. To this end, we observe that the below statements are equivalent:

- $\mathcal{M}, \sigma', n \models t_1(\psi_1 \mathbf{U} \psi_2)$
- $\mathcal{M}, \sigma', n \models t_1(\psi_1) \mathbf{U} t_1(\psi_2)$ (by definition of t_1)
- There is $n' \geq n$ such that $\mathcal{M}, \sigma', n' \models t_1(\psi_2)$ and for all $n \leq n'' < n'$, we have $\mathcal{M}, \sigma', n'' \models t_1(\psi_1)$ (by definition of \models)
- There is $n' \geq n$ such that $\mathcal{M}', (n', \sigma') \models \psi_2$ and for all $n \leq n'' < n'$, we have $\mathcal{M}', (n'', \sigma') \models \psi_1$ (by induction hypothesis)
- $\mathcal{M}', (n, \sigma') \models \psi_1 \mathbf{U} \psi_2$ (by definition of \models)

This allows us to show that $\mathcal{M}', (0, \sigma) \models \varphi$. \square

As $t_1(\varphi)$ is computed in logarithmic-space, we get the following.

Corollary 2. SLTL-satisfiability is ExpSpace-hard.

Note that this result contradicts the PSpace upper bound shown by Gigante et al. [20, Theorem 28], as PSpace and ExpSpace are known to be distinct complexity classes.

3.2 Translation from SLTL to PTLxS5

The translation from SLTL to PTLxS5 is (slightly) more complex since SLTL models interpret standpoint symbols, which have no natural counterpart in PTLxS5 models. As we will show, however, standpoints can be simulated in PTLxS5 with fresh propositional variables. This, in particular, requires an additional formula (called χ_n below) simulating the requirements that each standpoint in an SLTL model has at least one associated trace and that if a trace is assigned to a standpoint, it is so throughout the entire timeline. As we show, such requirements can be easily expressed in PTLxS5.

Given an SLTL formula φ , we construct a PTLxS5 formula $t_2(\varphi)$, by applying the translation map t_2 such that t_2 is the identity map for propositional variables, it is homomorphic for Boolean and temporal connectives, and the following hold for all $\mathbf{s}, \mathbf{s}' \in \mathcal{S}$:

$$\begin{aligned} t_2(\diamond_* \psi) &\stackrel{\text{def}}{=} \diamond t_2(\psi), & t_2(\diamond_{\mathbf{s}} \psi) &\stackrel{\text{def}}{=} \diamond(\mathbf{s} \wedge t_2(\psi)), \\ t_2(\square_* \psi) &\stackrel{\text{def}}{=} \square t_2(\psi), & t_2(\square_{\mathbf{s}} \psi) &\stackrel{\text{def}}{=} \square(\mathbf{s} \rightarrow t_2(\psi)), \\ & & t_2(\mathbf{s} \preceq \mathbf{s}') &\stackrel{\text{def}}{=} \square(\mathbf{s} \rightarrow \mathbf{s}'). \end{aligned}$$

Assuming that the standpoint symbols in φ are $\mathbf{s}_1, \dots, \mathbf{s}_n$, we let

$$\chi_n \stackrel{\text{def}}{=} \bigwedge_{1 \leq i \leq n} (\diamond \mathbf{s}_i) \wedge \square \bigwedge_{1 \leq i \leq n} (\mathbf{G} \mathbf{s}_i \vee \mathbf{G} \neg \mathbf{s}_i).$$

As we show next, checking satisfiability of φ in SLTL reduces to checking satisfiability of $\chi_n \wedge t_2(\varphi)$ in PTLxS5.

Lemma 3. φ is SLTL-satisfiable iff $\chi_n \wedge t_2(\varphi)$ is PTLxS5-satisfiable.

Proof sketch. Assume that φ is SLTL-satisfiable, so there are an SLTL model $\mathcal{M} = (\Pi, \lambda)$ and a trace $\sigma \in \Pi$ such that $\mathcal{M}, \sigma, 0 \models \varphi$. Let us build a PTLxS5 model $\mathcal{M}' = (\mathbb{N} \times W, L)$ such that $W \stackrel{\text{def}}{=} \Pi$

and $L((n, \sigma')) \stackrel{\text{def}}{=} \sigma'(n) \cup \{\mathbf{s} \mid \sigma' \in \lambda(\mathbf{s})\}$ for all $(n, \sigma') \in \mathbb{N} \times W$. We show that $\mathcal{M}', (0, \sigma) \models \chi_n \wedge t_2(\varphi)$. First, we observe that $\mathcal{M}', (0, \sigma) \models \chi_n$. This follows from the definition of χ_n and the fact that for all $\sigma' \in \Pi$ and for all $j \geq 0$, we have $\mathbf{s}_i \in L((j, \sigma'))$ iff $\sigma' \in \lambda(\mathbf{s}_i)$. Second, by structural induction, we can show that for all subformulae ψ of φ , all $\sigma' \in \Pi$, and all $j \in \mathbb{N}$, we get $\mathcal{M}, \sigma', j \models \psi$ iff $\mathcal{M}', (j, \sigma') \models t_2(\psi)$. Hence $\mathcal{M}', (0, \sigma) \models \chi_n \wedge t_2(\varphi)$.

Now, assume that there is a PTLxS5 model $\mathcal{M} = (\mathbb{N} \times W, L)$ and $(0, w) \in \mathbb{N} \times W$ such that $\mathcal{M}, (0, w) \models \chi_n \wedge t_2(\varphi)$. Let us build an SLTL model $\mathcal{M}' = (\Pi, \lambda)$ and a trace $\sigma \in \Pi$ such that $\mathcal{M}, \sigma, 0 \models \varphi$. As in the proof of Lemma 1, let Π consist of all traces

$$\sigma_w \stackrel{\text{def}}{=} L((0, w')), L((1, w')), L((2, w')), L((3, w')), \dots$$

with $w' \in W$. This time, the definition of λ exploits the assumption that $\mathcal{M}, (0, w) \models \chi_n$. In particular, to define λ we let $\sigma_{w'} \in \lambda(\mathbf{s}_i) \stackrel{\text{def}}{=} \mathcal{M}, (0, w') \models \mathbf{G} \mathbf{s}_i$, for all $i \in [1, n]$ and $w' \in W$. By the definition of λ and the form of χ_n , for each \mathbf{s}_i we have $\lambda(\mathbf{s}_i) \neq \emptyset$. This guarantees that \mathcal{M} is an SLTL model. By structural induction, we can show that for all subformulae ψ of φ and for all $(j, w') \in \mathbb{N} \times W$, we have $\mathcal{M}, (j, w') \models t_2(\psi)$ iff $\mathcal{M}', \sigma_{w'}, j \models \psi$. Hence, $\mathcal{M}', \sigma, 0 \models \varphi$ and so φ is satisfiable in SLTL. \square

As the construction of $\chi_n \wedge t_2(\varphi)$ is feasible in logarithmic-space, we obtain the following corollary.

Corollary 4. SLTL-satisfiability is in ExpSpace.

Together with Corollary 2 we obtain tight complexity bounds for SLTL, which is the main result of this section:

Theorem 5. SLTL-satisfiability is ExpSpace-complete.

Logarithmic-space reductions between SLTL and PTLxS5 emphasize how close are these formalisms, a property that remained unnoticed so far. This allows us to establish the ExpSpace-completeness of SLTL-satisfiability in a transparent way. Therefore, the analysis of the properties of the tableau-style calculus for SLTL designed by Gigante et al. [20] needs, in the best case, lead to the ExpSpace upper bound.

4 PSpace Fragment of SLTL

Corollary 2 can be viewed as a negative result for the usability of SLTL (it is of course positive in terms of knowledge about SLTL properties), but as shown below, there is some room to find interesting fragments that include both LTL and PSL, but can be decided in polynomial-space. Before presenting a fragment of SLTL in which satisfiability is PSpace-complete, however, we provide more intuitions behind ExpSpace-hardness of full SLTL. This is helpful to discard syntactic features that lead to high complexity.

4.1 Computationally challenging behaviour of SLTL

To obtain a better understanding of the ExpSpace-hardness of full SLTL and of the mismatch between our result in Section 3.1 and the PSpace bound stated by Gigante et al. [20, Theorem 28], we present a specific SLTL formula φ_C whose satisfiability is particularly hard to decide. This will also prove useful for constructing PSpace fragments of SLTL, as they should disallow features expressing the computationally challenging behaviour of φ_C .

Our construction of φ_C exploits the following LTL formula C_n , encoding a binary counter from 0 to $2^n - 1$, where the propositional

variables p_1, \dots, p_n correspond to consecutive bits of the counter with p_1 being the most significant bit:

$$\begin{aligned} & (\neg p_1 \wedge \dots \wedge \neg p_n) \wedge G(p_1 \wedge \dots \wedge p_n \rightarrow X(\neg p_1 \wedge \dots \wedge \neg p_n)) \\ & \wedge \bigwedge_{1 \leq i \leq n} G\left(\left(\neg p_i \wedge \bigwedge_{i < i' \leq n} p_{i'}\right) \rightarrow \left(\bigwedge_{i < i' \leq n} (X\neg p_{i'}) \wedge Xp_i \right.\right. \\ & \qquad \qquad \qquad \left.\left. \wedge \bigwedge_{1 \leq i' < i} (p_{i'} \leftrightarrow Xp_{i'})\right)\right). \end{aligned}$$

In every position i of an LTL trace σ , the propositional variables p_1, \dots, p_n encode the bits b_1, \dots, b_n (with b_1 being the most significant bit) such that $b_j = 1$ iff $p_j \in \sigma(i)$. We write $\sigma, i \models C = m$ if the counter has value m in the position i , that is $b_1 \dots b_n$ represents the number m . It is easy to see that if $\sigma, i \models C_n$, then the counter has value 0 in the position i and in every next position this value increases by 1 until the counter reaches the value $2^n - 1$. If this is the case, in the next position the value is 0 and the process of counting (modulo 2^n) repeats. Hence, $\sigma, i \models C = 0$ and for any $j \geq i$, we have $\sigma, j \models C = m'$ where $m' \equiv j - i \pmod{2^n}$.

We use C_n to define $\varphi_C \stackrel{\text{def}}{=} G(\diamond_s(C_n \wedge p \wedge XG\neg p))$ (of polynomial-size in n). As we show next, if $\mathcal{M}, \sigma, 0 \models \varphi_C$ for some SLTL model $\mathcal{M} = (\Pi, \lambda)$, then Π must contain infinitely many traces and for every position $i > 2^n$ there are exponentially many (at least 2^n) traces in Π which are pairwise different with respect to the valuation of p_1, \dots, p_n at the position i . The latter, in particular, seems to contradict the first paragraph of the proof of [20, Lemma 27] used to argue for PSpace satisfiability of SLTL.

Proposition 6. *The formula φ_C is SLTL satisfiable. Moreover, for each SLTL model $\mathcal{M} = (\Pi, \lambda)$ and $\sigma \in \Pi$ such that $\mathcal{M}, \sigma, 0 \models \varphi_C$, the following hold: (1) $\lambda(\mathbf{s})$ is infinite, (2) for each $i > 2^n$ and $m \in \{0, \dots, 2^n - 1\}$, there is $\sigma \in \Pi$ such that $\mathcal{M}, \sigma, i \models C = m$, and (3) for all positions $i > 2^n$, there are $\sigma_1, \dots, \sigma_{2^n}$ in Π such that $|\{\sigma_j(i) \cap \{p_1, \dots, p_n\} \mid j \in \{1, \dots, 2^n\}\}| = 2^n$.*

Proof. Assume that $\mathcal{M}, \sigma, 0 \models \varphi_C$. Firstly, let us show that $\lambda(\mathbf{s})$ is necessarily infinite. Since $\mathcal{M}, \sigma, 0 \models G(\diamond_s(C_n \wedge p \wedge XG\neg p))$, for each $i \geq 0$, there is a witness trace $\sigma_i^\dagger \in \lambda(\mathbf{s})$ such that $\mathcal{M}, \sigma_i^\dagger, i \models C_n \wedge p \wedge XG\neg p$. Hence, for all $i' > i$, we have $\mathcal{M}, \sigma_i^\dagger, i' \models \neg p$ and $\mathcal{M}, \sigma_{i'}^\dagger, i' \models p$, and so, σ_i^\dagger is distinct from $\sigma_{i'}^\dagger$. Consequently, $\{\sigma_0^\dagger, \sigma_1^\dagger, \dots\}$ is an infinite set of distinct traces, all belonging to $\lambda(\mathbf{s})$.

To show the second item in the proposition, let $i > 2^n$ and let $\sigma_1^\dagger, \dots, \sigma_{2^n}^\dagger$ be 2^n witness traces as defined above. So, for all $j \in \{1, \dots, 2^n\}$, we have $\mathcal{M}, \sigma_j^\dagger, j \models C_n$. Assuming that $\mathcal{M}, \sigma_1^\dagger, i \models C = m_1$ for some $m_1 \in \{0, \dots, 2^n - 1\}$, for all $j \in \{1, \dots, 2^n\}$ we have $\mathcal{M}, \sigma_j^\dagger, i \models C = m_j$ with $m_j \equiv m_1 + (j - 1) \pmod{2^n}$. Since m_1, \dots, m_{2^n} are pairwise distinct, we obtain also that $|\{\sigma_j^\dagger(i) \cap \{p_1, \dots, p_n\} \mid j \in \{1, \dots, 2^n\}\}| = 2^n$, and so, the third item from the proposition holds.

Let us conclude by showing that φ_C is satisfiable, as stated in the proposition. For this we can construct a model $\mathcal{M} = (\Pi, \lambda)$ with infinitely many traces $\Pi \stackrel{\text{def}}{=} \{\sigma_0, \sigma_1, \dots\}$ and with $\lambda(\mathbf{s}) \stackrel{\text{def}}{=} \Pi$. Each trace $\sigma_j \in \Pi$ is such that $\mathcal{M}, \sigma_j, j \models C_n \wedge p \wedge XG\neg p$. Note that such σ_j needs to exist and has a uniquely determined valuation of p, p_1, \dots, p_n in all positions not smaller than j , whereas their valuation in positions smaller than j is irrelevant for our construction. Then, by the form of φ_C , we get $\mathcal{M}, \sigma, 0 \models \varphi_C$. \square

Proposition 6 specifies the types of computationally hard formulae of SLTL. In forthcoming Section 4.2, we introduce a way of restricting SLTL, in which the behaviour of φ_C cannot be reconstructed, and which yields a PSpace-complete satisfiability problem.

4.2 Limiting the interplay between connectives

In this section we show that a PSpace upper bound for the satisfiability problem in SLTL can be obtained by limiting the interplay between standpoint and temporal operators. To this end, we focus on SLTL formulae which do not allow for LTL connectives occurring in the scope of standpoint modal operators; in the sequel we call this fragment LTL[PSL]. By way of example, the formula $\Box_*(G\neg Malf \rightarrow Test)$ from Section 2.1 does not belong to LTL[PSL]. However the following is already an LTL[PSL] formula: $G(\Box_*(\neg Malf) \rightarrow \Box_* Test)$, which states that if always in future all countries agree that a medical device does not malfunction, then all these countries agree that this device is safe according to testing. It is worth observing that the formula φ_C from Section 4.1, illustrating computationally challenging behaviour, does not belong to LTL[PSL] (because of XG in the scope of \diamond_s). In contrast, any formula of the form $G\Box_*\varphi_1 \wedge G(\varphi_2 \rightarrow X\varphi_3)$ where $\varphi_1, \varphi_2, \varphi_3$ are PSL formulae (see Section 2.2) are in LTL[PSL].

As we show in the remaining part of this section, we can check satisfiability of LTL[PSL] formulae in PSpace by adapting the Büchi automaton construction developed for LTL [5, 15] and by performing model-theoretical constructions based on a refinement of the finite model property for PSL (see Theorem 9).

Let us fix an LTL[PSL] formula φ and let $\mathcal{P}(\varphi)$ be the set of propositional variables occurring in it. Let I (for ‘inequality’) be the set of subformulae of φ of the form $\mathbf{s} \preceq \mathbf{s}'$. To check satisfiability of φ , our procedure considers all possible partitions $D = (I^+, I^-)$ of I into sets I^+ and I^- ; intuitively such a partition states that formulae in I^+ are true and those in I^- are false. Since $\mathbf{s} \preceq \mathbf{s}'$ is a global statement (either it is satisfied in all traces and all positions of a model, or it is not satisfied in any of them), its truth value can be nondeterministically guessed from the very beginning. Moreover, such a guess is feasible in PSpace since $\text{NPSpace} = \text{PSpace}$ [30]. Given $D = (I^+, I^-)$, we encode the above intuition with the formula

$$\chi_D \stackrel{\text{def}}{=} G\left(\bigwedge_{(\mathbf{s} \preceq \mathbf{s}') \in I^+} (\mathbf{s} \preceq \mathbf{s}') \wedge \bigwedge_{(\mathbf{s} \preceq \mathbf{s}') \in I^-} (\diamond_s p_{\mathbf{s}, \mathbf{s}'} \wedge \neg \diamond_{\mathbf{s}'} p_{\mathbf{s}, \mathbf{s}'})\right)$$

where each $p_{\mathbf{s}, \mathbf{s}'}$ is a fresh propositional variable. This can be done as \mathcal{P} is countably infinite. Note that χ_D encodes that each $\mathbf{s} \preceq \mathbf{s}'$ in I^+ holds true always in future and that each $\mathbf{s} \preceq \mathbf{s}'$ in I^- is always false. We also define $\varphi_D \stackrel{\text{def}}{=} \varphi[I^+ \mapsto \top, I^- \mapsto \perp] \wedge \chi_D$, where $\varphi[I^+ \mapsto \top, I^- \mapsto \perp]$ is the formula obtained from φ by replacing each $(\mathbf{s} \preceq \mathbf{s}') \in I^+$ with \top and each $(\mathbf{s} \preceq \mathbf{s}') \in I^-$ with \perp . Note that χ_D is an LTL[PSL] formula.

The purpose of the construction of φ_D , leading to Lemma 7, is to nondeterministically choose which inequalities $\mathbf{s} \preceq \mathbf{s}'$ hold, and in particular to enforce the non-satisfaction of some $\mathbf{s} \preceq \mathbf{s}'$ by the existence of two traces, see the formula χ_D . Notice that $\mathbf{s} \preceq \mathbf{s}'$ occurs only positively in χ_D . In forthcoming Theorem 9, we show how to construct PSL models based on φ_1 , defined as a conjunction of inequalities of the form $\mathbf{s} \preceq \mathbf{s}'$. This step is therefore instrumental to handle the inequalities $\mathbf{s} \preceq \mathbf{s}'$.

Lemma 7. *Satisfiability of φ reduces to checking if there exists a partition $D = (I^+, I^-)$ of I such that φ_D is satisfiable.*

Therefore, in what follows we will consider an arbitrary partition $D = (I^+, I^-)$ of I , and focus on checking satisfiability of φ_D .

We write $\text{cl}(\varphi_D)$ to denote the *closure set* of φ_D defined as the smallest set containing all the subformulae ψ of φ_D as well as \perp and \top , which is closed under taking negations (as usually, we do not

allow for double negations by identifying $\neg\neg\psi$ with ψ) and such that $\psi\mathbf{U}\psi' \in \text{cl}(\varphi_D)$ implies $\mathbf{X}(\psi\mathbf{U}\psi') \in \text{cl}(\varphi_D)$. We can observe that $|\text{cl}(\varphi_D)| \leq 4 \cdot |\varphi_D|$ and the number of formulae in the set $\text{cl}(\varphi_D)$, whose outermost connective is a modality of the form \diamond_s or \square_s , is bounded by $|\varphi_D|$.

As in the standard automaton construction for LTL [5, 15, 16], we call a set $B \subseteq \text{cl}(\varphi)$ *maximally consistent* if it satisfies the properties:

- $\top \in B$ and $\perp \notin B$; $\psi \in B$ iff $\neg\psi \notin B$, for all $\neg\psi \in \text{cl}(\varphi_D)$,
- $\psi_1 \wedge \psi_2 \in B$ iff $\psi_1 \in B$ and $\psi_2 \in B$, for all $\psi_1 \wedge \psi_2 \in \text{cl}(\varphi_D)$,
- $\psi_1 \mathbf{U}\psi_2 \in B$ iff $\psi_2 \in B$ or $\{\psi_1, \mathbf{X}(\psi_1 \mathbf{U}\psi_2)\} \subseteq B$, for all $\psi_1 \mathbf{U}\psi_2 \in \text{cl}(\varphi_D)$.

Moreover, we introduce an additional property specific to SLTL; we say that B is *standpoint-consistent* if the PSL formula $\bigwedge_{\psi \in B \cap \text{PSL}} \psi$ (we treat here PSL as the set of all well-formed PSL formulae) is satisfiable in standpoint logic PSL. It is worth noting that checking standpoint-consistency is decidable and, in particular, NP-complete [22, Corollary 3] as we have discussed in Section 2.1 (see also the forthcoming Theorem 9).

We call $B \subseteq \text{cl}(\varphi_D)$ *s-elementary* if it is maximally consistent and standpoint-consistent. We will use *s-elementary* sets as states of an automaton.

We are ready now to define an automaton \mathcal{A}_{φ_D} which we use for checking satisfiability of φ_D . We let \mathcal{A}_{φ_D} be a generalised non-deterministic Büchi automaton $\mathcal{A}_{\varphi_D} = (\mathcal{Q}, 2^{\mathcal{P}(\varphi_D)}, \delta, \mathcal{Q}_0, \mathcal{F})$, whose components are as follows:

- the set of states, \mathcal{Q} , consists of all s-elementary sets $B \subseteq \text{cl}(\varphi_D)$,
- the automaton alphabet is the powerset $2^{\mathcal{P}(\varphi_D)}$,
- the set of initial states is $\mathcal{Q}_0 = \{B \in \mathcal{Q} \mid \varphi_D \in B\}$,
- the set of accepting sets, \mathcal{F} , is the family containing for each $\psi_1 \mathbf{U}\psi_2 \in \text{cl}(\varphi_D)$ the set $\{B \in \mathcal{Q} \mid \psi_1 \mathbf{U}\psi_2 \notin B \text{ or } \psi_2 \in B\}$,
- the transition relation $\delta : \mathcal{Q} \times 2^{\mathcal{P}(\varphi_D)} \rightarrow 2^{\mathcal{Q}}$ is such that $\delta(B, A) = \emptyset$ if $A \neq B \cap \mathcal{P}(\varphi_D)$, and otherwise $\delta(B, A)$ is the set of all s-elementary $B' \subseteq \text{cl}(\varphi_D)$ which satisfy the following condition: for each $\mathbf{X}\psi \in \text{cl}(\varphi_D)$, we have $\mathbf{X}\psi \in B'$ iff $\psi \in B'$.

In what follows we show that accepting runs of \mathcal{A}_{φ_D} correspond to SLTL models of φ_D , and so, satisfiability of φ_D is equivalent to nonemptiness of \mathcal{A}_{φ_D} language. In the next lemma we show the first implication of this equivalence.

Lemma 8. *If φ_D is SLTL-satisfiable, then the language of \mathcal{A}_{φ_D} is non-empty.*

Proof sketch. Assume that $\mathcal{M}, \sigma, 0 \models \varphi_D$, for some SLTL model $\mathcal{M} = (\Pi, \lambda)$ and $\sigma \in \Pi$. We let states B_0, B_1, \dots be such that each $B_i = \{\psi \in \text{cl}(\varphi_D) \mid \mathcal{M}, \sigma, i \models \psi\}$ and we let input word $A_0 A_1 \dots$ be such that each $A_i = B_i \cap \mathcal{P}(\varphi_D)$. As we will show, B_0, B_1, \dots is a run of \mathcal{A}_{φ_D} on the word $A_0 A_1 \dots$, and this run is accepting. For the former statement, it suffices to observe that, by the definition, each B_i is s-elementary and $B_{i+1} \in \delta(B_i, A_i)$. To prove that the run B_0, B_1, \dots is accepting, we can show that for each accepting set $F \in \mathcal{F}$, there are infinitely many i with $B_i \in F$. This follows from the standard argument used in the automaton construction for LTL [5, 15], since the accepting sets \mathcal{F} in our construction are defined in the same way as in the case of LTL. As B_0, B_1, \dots is an accepting run of \mathcal{A}_{φ_D} , the language of \mathcal{A}_{φ_D} is non-empty. \square

Next, we will show that the converse of Lemma 8 (forthcoming Lemma 10) holds true. This direction is more complex and requires establishing new properties for PSL. Given an accepting run B_0, B_1, \dots of \mathcal{A}_{φ_D} , we aim to construct an SLTL model of

φ_D . By standpoint-consistency of the B_i 's, there exists a sequence $(\mathcal{M}_0, \pi_0), (\mathcal{M}_1, \pi_1), \dots$ of pointed PSL models (i.e. pairs consisting of a PSL model and one of its precisifications) such that for all $i \geq 0$, we have $\mathcal{M}_i, \pi_i \models \bigwedge_{\psi \in B_i \cap \text{PSL}} \psi$. From such a sequence, we aim to construct an SLTL model $\mathcal{M} = (\Pi, \lambda)$ and a trace $\sigma \in \Pi$ such that $\mathcal{M}, \sigma \models \varphi_D$. The main challenge is to construct the set Π of traces in \mathcal{M} from the sets Π_0, Π_1, \dots of precisifications in models $\mathcal{M}_0, \mathcal{M}_1, \dots$. This task is reminiscent of known constructions of models from partial models, for instance, as developed in the mosaic method for temporal logics [25], see also [33]. However, in our case, the construction requires exploiting properties specific to PSL.

To address this challenge, we show a new model-theoretic result for PSL, which is interesting on its own as, for example, it implies NP-membership of PSL-satisfiability. Our result states that the satisfiability of PSL formulae can be witnessed by PSL models of a specific form and size, so it establishes a “normalised small model property” for PSL. We show the result for PSL formulae which are relevant for our construction. In particular, for formulae of the form $\varphi_1 \wedge \varphi_2$ such that φ_1 is a conjunction of formulae $\mathbf{s} \preceq \mathbf{s}'$ and φ_2 is a PSL formula in negation normal form with no occurrences of $\mathbf{s} \preceq \mathbf{s}'$ (see the formula φ_D involved in Lemma 7). Without any loss of generality, we can assume that the universal standpoint $*$ occurs in φ_1 (indeed, we can always add to φ_1 the formula $* \preceq *$, as it is equivalent to \top).

Our normalised small model property states that satisfiability of $\varphi_1 \wedge \varphi_2$ implies existence of some PSL model of a specific form. This model has precisifications represented by pairs (S, j) such that j is a natural number and S belongs to a set \mathbb{S} defined as follows (based on the form of φ_1). We let R be the reflexive and transitive closure of the relation $\{(\mathbf{s}, \mathbf{s}') \mid \mathbf{s} \preceq \mathbf{s}' \text{ occurs in } \varphi_1\} \cup \{(\mathbf{s}, *) \mid \mathbf{s} \in \mathcal{S}(\varphi_1 \wedge \varphi_2)\}$. Hence, if $(\mathbf{s}, \mathbf{s}') \in R$, then φ_1 entails $\mathbf{s} \preceq \mathbf{s}'$. We let $R(\mathbf{s})$ be the set of all \mathbf{s}' such that $(\mathbf{s}, \mathbf{s}') \in R$. Then we define the set $\mathbb{S} \stackrel{\text{def}}{=} \{R(\mathbf{s}) \mid \mathbf{s} \in \mathcal{S}(\varphi_1 \wedge \varphi_2)\}$. With these symbols in hand we are ready to formulate the small model property of PSL.

Theorem 9. *Let $\varphi_1 \wedge \varphi_2$ be a PSL-formula of the form described above, which mentions N_1 standpoint symbols and N_2 occurrences of \diamond . If $\varphi_1 \wedge \varphi_2$ is PSL-satisfiable, then for every natural number $N > N_1 + N_2$ there is a PSL model $\mathcal{M}^* = (\Pi^*, V^*)$ such that:*

1. $\Pi^* = \mathbb{S} \times \{1, \dots, N\}$,
2. $\mathcal{M}^*, (S^*, 1) \models \varphi_1 \wedge \varphi_2$ with $S^* = R(*)$, and
3. for each $(S, j) \in \Pi^*$, it holds that $S = \{\mathbf{s} \in \mathcal{S}(\varphi_1 \wedge \varphi_2) \mid (S, j) \in V^*(\mathbf{s})\}$.

Proof sketch. The proof employs principles first used to show the small model property for S5 [12, Chapter 6], for counting logics [26, Chapter 1], and for PSL itself [21, Section 4.4].

Given a PSL model \mathcal{M}^\dagger and a precisification π^\dagger such that $\mathcal{M}^\dagger, \pi^\dagger \models \varphi_1 \wedge \varphi_2$, we perform four transformations to modify $\mathcal{M}^\dagger, \pi^\dagger$ into \mathcal{M}^*, π^* satisfying Statements 1–3 from the theorem, where $\pi^* = (S^*, 1)$. The steps are schematised below.

$$\mathcal{M}^\dagger, \pi^\dagger \xrightarrow{\text{Elect}} \mathcal{M}, \pi \xrightarrow{\text{Select}} \mathcal{M}^f, \pi^f \xrightarrow{\text{Normalise}} \mathcal{M}^{fn}, \pi^{fn} \xrightarrow{\text{Populate}} \mathcal{M}^*, \pi^*.$$

Let us describe briefly the objectives of each reduction.

(Elect) The goal of transforming $\mathcal{M}^\dagger, \pi^\dagger$ into \mathcal{M}, π is to guarantee that the standpoint symbols labelling π are exactly those in S^* . At most one precisification is added to \mathcal{M}^\dagger to obtain \mathcal{M} .

(Select) The construction of \mathcal{M}^f, π^f from \mathcal{M}, π amounts to select a *finite* subset of precisifications from \mathcal{M} to witness the satisfaction of all the \diamond -formulae. This step is analogous to the way the small model property is shown for the modal logic S5.

(Normalise) The construction of $\mathcal{M}^{fn}, \pi^{fn}$ from \mathcal{M}^f, π^f guarantees that for any precisification π in \mathcal{M}^{fn} , the set of standpoint symbols whose valuation contains π , belongs to \mathbb{S} . To do so, precisifications are “copied”, preserving the satisfaction of propositional variables but possibly updating the valuation of standpoints.

(Populate) This step to get \mathcal{M}^*, π^* from $\mathcal{M}^{fn}, \pi^{fn}$ consists in “copying” precisifications, so that the set of precisifications can be identified with $\mathbb{S} \times [1, N]$.

It can be shown that \mathcal{M}^*, π^* constructed in this way is possible and satisfies all Statements 1–3 from the theorem. \square

Interestingly, as a consequence of Theorem 9, we obtain the NP-membership for PSL-satisfiability. Indeed, given a PSL formula φ , it suffices to non-deterministically guess a partition $D = (I^+, I^-)$ of formulae $\mathbf{s} \preceq \mathbf{s}'$ in φ , and to verify satisfiability of $\bigwedge_{(\mathbf{s} \preceq \mathbf{s}') \in I^+} (\mathbf{s} \preceq \mathbf{s}') \wedge \bigwedge_{(\mathbf{s} \preceq \mathbf{s}') \in I^-} (\diamond_{\mathbf{s}} p_{\mathbf{s}, \mathbf{s}'} \wedge \neg \diamond_{\mathbf{s}'} p_{\mathbf{s}, \mathbf{s}'}) \wedge \varphi[I^+ \mapsto \top, I^- \mapsto \perp]$. The form of the obtained formula allows us to apply Theorem 9. Hence, to check satisfiability, it remains to guess a model with $|\mathbb{S}|(N_1 + N_2 + 1)$ precisifications. The whole procedure is in NP. Above all, Theorem 9 is a key to show the converse of Lemma 8, stated below.

Lemma 10. *If the language of \mathcal{A}_{φ_D} is not non-empty, then φ_D is SLTL-satisfiable*

Proof sketch. Assume that B_0, B_1, \dots is an accepting run of \mathcal{A}_{φ_D} on a word $A_0 A_1 \dots$. We construct an SLTL model $\mathcal{M} = (\Pi, \lambda)$ and a trace $\sigma \in \Pi$ such that $\mathcal{M}, \sigma, 0 \models \varphi_D$. For each $i \in \mathbb{N}$, by standpoint consistency of B_i , there exists a PSL model $\mathcal{M}_i = (\Pi_i, V_i)$ and a precisification $\pi_i \in \Pi_i$ such that $\mathcal{M}_i, \pi_i \models \bigwedge_{\psi \in B_i \cap \text{PSL}} \psi$. Each formula $\bigwedge_{\psi \in B_i \cap \text{PSL}} \psi$ can be written in the form $\varphi_1 \wedge \varphi_2$ assumed in Theorem 9. Indeed, the only atomic formulae of the form $\mathbf{s} \preceq \mathbf{s}'$ in $\text{cl}(\varphi_D)$ are those from I^+ , and due to χ_D , each $\mathbf{s} \preceq \mathbf{s}'$ can occur only positively in B_i (otherwise $\bigwedge_{\psi \in B_i \cap \text{PSL}} \psi$ is not satisfiable). For each $i \in \mathbb{N}$, the formula $\bigwedge_{\psi \in B_i \cap \text{PSL}} \psi$ yields R, N_1 , and N_2 used in Theorem 9. Importantly, all these R are the same, all $N_1 \leq |\mathcal{S}(\varphi_D)|$, and all $N_2 \leq |\varphi_D|^2$. Therefore we can apply Theorem 9 with $N = |\mathcal{S}(\varphi_D)| + |\varphi_D|^2 + 1$ to obtain that there are PSL models $\mathcal{M}'_i = (\Pi'_i, V'_i)$ such that $\Pi'_i = \mathbb{S} \times \{1, \dots, N\}$, $\mathcal{M}'_i, (S^*, 1) \models \bigwedge_{\psi \in B_i \cap \text{PSL}} \psi$ (with $S^* = R(*)$), and for all $(S, j) \in \Pi'_i$, we have $\{\mathbf{s} \in \mathcal{S}(\varphi_D) \mid (S, j) \in V'_i(\mathbf{s})\} = S$. We use these models to define an SLTL model $\mathcal{M} = (\Pi, \lambda)$ and $\sigma \in \Pi$. We let Π be $\{\sigma_{(S, j)} \mid (S, j) \in \mathbb{S} \times \{1, \dots, N\}\}$ and λ be such that

- for all $\mathbf{s} \in \mathcal{S}(\varphi_D)$, $\sigma_{(S, j)} \in \lambda(\mathbf{s})$ iff $\mathbf{s} \in S$,
- for all $(S, j) \in \mathbb{S} \times \{1, \dots, N\}$, for all $k \geq 0$, we have $\sigma_{(S, j)}(k) \stackrel{\text{def}}{=} \{p \in \mathcal{P}(\varphi_D) \mid (S, j) \in V'_k(p)\}$.

Finally, we let σ be $\sigma_{(S^*, 1)}$. The construction of \mathcal{M} is schematised in Figure 1. Observe that $\varphi_D \in B_0$. Hence, to prove that $\mathcal{M}, \sigma_{(S^*, 1)}, 0 \models \varphi_D$, it suffices to show by structural induction that for all subformulae ψ of φ_D and all $i \in \mathbb{N}$ we have $\psi \in B_i$ iff $\mathcal{M}, \sigma_{(S^*, 1)}, i \models \psi$. In the basis of the induction we let ψ be any subformula which does not mention LTL connectives. Note that for such ψ , the equivalence follows from the construction of \mathcal{M}'_i . Regarding the inductive step, since φ does not mention LTL connectives in the scope of standpoint operators, the cases in the inductive steps are for Boolean and LTL connectives only. Thus, the proof is analogous as in the case of automaton construction for LTL [5, 15]. \square

Lemmas 8 and 10 allow us to reduce the satisfiability of φ_D to checking the nonemptiness of \mathcal{A}_{φ_D} language. This, as we show next, leads to tight PSpace complexity bound for satisfiability checking.

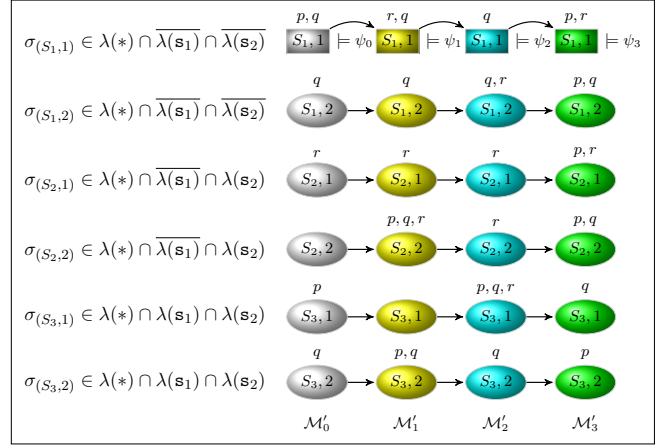


Figure 1. Construction of \mathcal{M} from \mathcal{M}'_i 's with $N = 2$, $S^* = S_1 = \{*\}$, $S_2 = \{*, s_2\}$, $S_3 = \{*, s_2, s_1\}$, and $\psi_i = \bigwedge_{\psi \in B_i \cap \text{PSL}} \psi$

Theorem 11. *LTL[PSL]-satisfiability problem is PSpace-complete.*

Proof. The lower bound is from the fact that LTL is a syntactic fragment of LTL[PSL] and is already PSpace-hard [31, Theorem 4.1]. For the upper bound, assume that we want to check if φ is satisfiable. Our procedure starts by guessing a partition D of subformulae of φ of the form $\mathbf{s} \preceq \mathbf{s}'$ (in nondeterministic polynomial-time), and constructing the formula φ_D . By Lemma 7, it remains to check if φ_D is satisfiable. This, however, by Lemmas 8 and 10, reduces to checking if the language of \mathcal{A}_{φ_D} is non-empty. The size (number of states) of \mathcal{A}_{φ_D} is exponentially large, but similarly as in the automata construction for LTL, we can use an “on the fly” approach to check in PSpace non-emptiness of \mathcal{A}_{φ_D} language [5, 15]. The difference between our procedure and the standard one for LTL is that we need to check if each constructed ‘on the fly’ state B_i of the automaton is standpoint-consistent. This, however, is feasible in NP due to NP-completeness of the satisfiability problem for PSL [22]. \square

As a corollary of the proof, we obtain also the following result.

Corollary 12. *If an LTL[PSL] formula φ is satisfiable, it is satisfied in an SLTL model with polynomially many (in the size of φ) traces.*

5 Concluding Remarks

We studied the computational properties of standpoint linear temporal logic SLTL. First, we proved that its satisfiability problem is ExpSpace-complete, contrary to the PSpace bound claimed recently in [20, Theorem 28]. To show this result, we designed reductions between SLTL and the multi-dimensional modal logic PTLxS5. Furthermore, we proposed a fragment of SLTL which has PSpace-complete satisfiability problem, as LTL [20]. This fragment disallows occurrences of temporal connectives in the scope of standpoint connectives; to show that the satisfiability problem for its formulae is in PSpace, we followed the automata-based approach [32] (similar to the well-known technique for LTL) but for which correctness requires to prove a new model-theoretic property about PSL (Theorem 9) that we find interesting for its own sake. In future it would be also interesting to implement practical decision procedures for SLTL and for the PSpace fragment we have introduced, apart from studying its expressive power.

Acknowledgements

We thank the referees for suggestions that help us to improve the quality of the document. Special thanks to the referee who pointed us to [8]. Przemysław A Wałęga was supported by the EPSRC projects OASIS (EP/S032347/1), ConCuR (EP/V050869/1) and UK FIRES (EP/S019111/1), as well as SIRIUS Centre for Scalable Data Access and Samsung Research UK.

References

- [1] G. Allwein and W. Harrison. Partially-ordered modalities. In *AiML'10*, pages 1–21. College Publications, 2010.
- [2] B. Aminof, G. De Giacomo, A. Murano, and S. Rubin. Planning under LTL environment specifications. In *ICAPS'19*, pages 31–39. AAAI Press, 2019.
- [3] F. Baader, S. Ghilardi, and C. Lutz. LTL over description logic axioms. *ACM Transactions on Computational Logic*, 13(3):21, 2012.
- [4] F. Baader, I. Horrocks, C. Lutz, and U. Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.
- [5] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT press, 2008.
- [6] M. Baldoni, L. Giordano, and A. Martelli. A tableau calculus for multimodal logics and some (un)decidability results. In *TABLEAUX'98*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 44–59. Springer, 1998.
- [7] M. Banerjee, M. Chakraborty, and A. Szalas. Logics from rough sets. *Journal of Applied Non-Classical Logics*, pages 1–3, 2024.
- [8] A. Barrière, B. Maubert, A. Murano, and S. Rubin. Reasoning about changes of observational power in logics of knowledge and time. In *AAMAS'19*, pages 971–979. International Foundation for Autonomous Agents and Multiagent Systems, 2019.
- [9] B. Bennett. Modal semantics for knowledge bases dealing with vague concepts. In *KR'98*, pages 234–244. Morgan Kaufmann, 1998.
- [10] B. Bennett. A theory of vague adjectives grounded in relevant observables. In *KR'06*, pages 36–45. AAAI Press, 2006.
- [11] B. Bennett. Standpoint semantics: a framework for formalising the variable meaning of vague terms. In *Understanding Vagueness — Logical, Philosophical and Linguistic Perspective*, pages 261–278. College Publications, 2011.
- [12] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
- [13] D. Calvanese, G. De Giacomo, and M. Vardi. Reasoning about actions and planning in LTL action theories. In *KR'02*, pages 593–602, 2002.
- [14] S. Demri. The complexity of regularity in grammar logics and related modal logics. *Journal of Logic and Computation*, 11(6):933–960, 2001.
- [15] S. Demri, V. Goranko, and M. Lange. *Temporal Logics In Computer Science: Finite-State Systems*, volume 58. Cambridge University Press, 2016.
- [16] J. Esparza and M. Blondin. *Automata Theory*. MIT Press, 2023.
- [17] L. Fariñas del Cerro and M. Penttonen. Grammar logics. *Logique et Analyse*, 121–122:123–134, 1988.
- [18] K. Fine. Vagueness, truth and logic. *Synthese*, pages 265–300, 1975.
- [19] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyashev. *Many-dimensional modal logics: theory and practice*. Cambridge University Press, 2003.
- [20] N. Gigante, L. Gómez Álvarez, and T. Lyon. Standpoint linear temporal logic. In *KR'23*, pages 311–321, 2023.
- [21] L. Gómez Álvarez. *Standpoint Logic: A Logic for Handling Semantic Variability, with Applications to Forestry Information*. PhD thesis, University of Leeds, School of Computing, 2019.
- [22] L. Gómez Álvarez and S. Rudolph. Standpoint logic: Multi-perspective knowledge representation. In *FOIS'22*, volume 344, pages 3–17, 2022.
- [23] L. Gómez Álvarez, S. Rudolph, and H. Strass. Tractable diversity: Scalable multiperspective ontology management via standpoint EL. In *IJ-CAI'23*, pages 3258–3267. ijcai.org, 2023.
- [24] L. Gómez Álvarez, S. Rudolph, and H. Strass. Pushing the boundaries of tractable multiperspective reasoning: A deduction calculus for standpoint EL+. In *KR'23*, pages 333–343, 2023.
- [25] M. Marx, S. Mikulas, and M. Reynolds. The mosaic method for temporal logics. In *TABLEAUX'00*, volume 1847 of *Lecture Notes in Artificial Intelligence*, pages 324–340. Springer, 2000.
- [26] S. Mikulás. *Taming logics*. PhD thesis, ILLC, 1995.
- [27] E. Orłowska. Reasoning with incomplete information: rough set based information logics. In *Incompleteness and Uncertainty in Information Systems*, pages 16–33. Springer, October 1994.
- [28] Z. Pawlak. Rough sets. *International Journal of Information and Computer Sciences*, 11:341–356, 1982.
- [29] A. Pnueli. The temporal logic of programs. In *FOCS'77*, pages 46–57. IEEE Computer Society Press, 1977.
- [30] W. Savitch. Relationships between nondeterministic and deterministic tape complexities. *JCSS*, 4(2):177–192, 1970.
- [31] A. Sistla and E. Clarke. The complexity of propositional linear temporal logic. *Journal of the ACM*, 32(3):733–749, 1985.
- [32] M. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115:1–37, 1994.
- [33] F. Wolter and M. Zakharyashev. On the decidability of description logics with modal operators. In *KR'98*, pages 512–523. Morgan Kaufmann, 1998.
- [34] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
- [35] L. Zadeh. Fuzzy logic and approximate reasoning. *Synthese*, 30:407–428, 1975.