Model Checking Linear Temporal Logic with Standpoint Modalities

Rajab Aghamov[1], **Christel Baier**[1], **Toghrul Karimov**[2], **Rupak Majumdar**[2], **Joël Ouaknine**[2], **Jakob Piribauer**[1], **Timm Spork**[1]

[1]Technische Universität Dresden, Dresden, Germany
[2]Max Planck Institute for Software Systems, Saarbrücken, Germany
{rajab.aghamov, christel.baier}@tu-dresden.de, {toghs, rupak, joel}@mpi-sws.org, jakob.piribauer@tu-dresden.de, timm.spork@tu-dresden.de

Abstract

Standpoint linear temporal logic (SLTL) is a recently introduced extension of linear temporal logic (LTL) with standpoint modalities. Intuitively, these modalities allow to express that, from agent a's standpoint, it is conceivable that a given formula holds. Besides the standard interpretation of the standpoint modalities, we introduce four new semantics, which differ in the information an agent can extract from the history. We provide a general model checking algorithm applicable to SLTL under any of the five semantics. Further we analyze the computational complexity of the corresponding model checking problems, obtaining PSPACE-completeness in three cases, which stands in contrast to the known EXPSPACE-completeness of the SLTL satisfiability problem.

1 Introduction

Automated reasoning about the dynamics of scenarios in which multiple agents with access to different information interact is a key problem in artificial intelligence and formal verification. Epistemic temporal logics are prominent, expressive formalisms to specify properties of such scenarios (see, e.g., (Halpern and Vardi 1986; Halpern 1986; Fagin et al. 2004; Bozzelli, Maubert, and Murano 2024)). The resulting algorithmic problems, however, often have non-elementary complexity or are even undecidable (see, e.g., (van der Meyden and Shilov 1999; Dima 2009; Bozzelli, Maubert, and Murano 2024)).

Aiming to balance expressiveness and computational tractability, (Álvarez and Rudolph 2021; Alvarez 2020) defines static *standpoint logics* that extend propositional logic with modalities $\langle\!\langle a \rangle\!\rangle \varphi$ expressing that "according to agent a, it is conceivable that φ" and the dual modalities $[\![a]\!]\varphi$ expressing that "according to a, it is unequivocal that φ". Standpoint logics and their extensions have proven useful to, e.g., reason about inconsistent formalizations of concepts in the medical domain to align different ontologies and in a forestry application, where different sources disagree about the global extent of forests (Álvarez and Rudolph 2021; Alvarez 2020; Alvarez, Rudolph, and Strass 2022).

Recently introduced combinations of linear temporal logic (LTL) with standpoint modalities (Gigante, Gómez Alvarez, and Lyon 2023; Demri and Walega 2024) enable reasoning about dynamical aspects of multi-agent systems. Enriching standpoint logic with a temporal dimension is essential for applications, e.g., in the verification of network and communication protocols or distributed systems. The focus of (Gigante, Gómez Alvarez, and Lyon 2023; Demri and Walega 2024) is the satisfiability problem for the resulting *standpoint LTL* (SLTL). In this paper, we consider the model-checking problem that asks whether all executions of a transition system satisfy a given SLTL-formula. To the best of our knowledge, this problem has not been addressed in the literature.

Whether the formula $\langle\!\langle a \rangle\!\rangle \varphi$ holds after a finite history, that is, whether agent a finds it plausible that φ holds in the future, depends on a's standpoint and on what was observable to a in the past. To illustrate this, consider a situation in which different political agents have different perceptions of how actions taken by the state influence future developments.These perceptions are the standpoints of the agents. After a series of events (i.e., a history), agents consider different future developments possible, depending on their standpoint and the parts of the history they are aware of. When reasoning about each other's standpoints, information might be exchanged in various ways. For example, during a discussion, agents may uncover previously unknown parts of the history or choose to disregard others' observations.

Besides the semantics for SLTL proposed in (Gigante, Gómez Alvarez, and Lyon 2023; Demri and Walega 2024), we introduce four additional semantics that differ in the amount of information agent a can access from the history and how information is transferred between agents. To formalize these semantics, we use the natural standpoint-logic approach with separate transition systems $\mathscr{T}_a$ describing the executions that are consistent with a's standpoint as in (Gigante, Gómez Alvarez, and Lyon 2023; Demri and Walega 2024), together with a main transition system $\mathscr{T}$ modeling the actual system. Unlike (Gigante, Gómez Alvarez, and Lyon 2023; Demri and Walega 2024), we assume the labels of the states in $\mathscr{T}_a$ to be from a subset P_a of the set P of atomic propositions of $\mathscr{T}$. The difference from the use of indistinguishability relations $\sim_a$ over states of one transition system for all agents $a \in Ag$, common in epistemic temporal logics, is that in LTLK they may be arbitrary equivalence relations unrelated to the atomic propositions, making the embedding of LTLK (without common knowledge) into SLTL challenging, despite the reverse being straightforward.

Under all five semantics, the intuitive meaning of $\langle\!\langle a \rangle\!\rangle \varphi$ is that there is a state s in $\mathscr{T}_a$, which is one of the potential current states from agent a's view of the history, and a path π in $\mathscr{T}_a$ from s such that π satisfies φ when a makes nondeterministic guesses for truth values of the atomic propositions outside P_a. Informally, the different semantics are:

– The *step semantics* $\models_{\text{step}}$ agrees with the semantics proposed in (Gigante, Gómez Alvarez, and Lyon 2023; Demri and Walega 2024). It assumes that only the number of steps performed in the past are accessible to the agents.

– The *pure observation-based semantics* $\models_{\text{pobs}}$ is in the spirit of the perfect-recall LTLK (LTL extended with knowledge operators) semantics of (Bozzelli, Maubert, and Murano 2024) where a can access exactly the truth values of the atomic propositions in P_a from the history. We provide an embedding of the pure observation-based semantics into LTLK. So, the same application domains as for LTLK such as multi-agent systems (Fagin et al. 2004) are applicable in our setting. The restriction imposed by SLTL structures that the indistinguishability relation of each agent a is given by a set of atomic proposition P_a in contrast to an arbitrary indistinguishability relation in LTLK structures is quite natural.

–The *public-history semantics* $\models_{\text{public}}$ can be seen as a perfect-recall variant of the LTLK semantics where all agents have full access to the history. It models agents sharing a knowledge base but using different future-prediction policies. This framework naturally extends to settings where agents share identical observations, such as a comprehensive medical dataset on the COVID-19 pandemic, yet adopt radically different interpretations of the data. For example, one agent might interpret epidemiological statistics and clinical reports as evidence that COVID-19 is a hoax, while another uses scientifically grounded inference rules on the same data to assess the pandemic's real-world impacts. Public-history semantics thus makes it possible to reason about both common data access and heterogeneous prediction policies within a single unified model.

– The *incremental semantics* $\models_{\text{incr}}$ is as $\models_{\text{decr}}$, but under the assumption that standpoint subformulas $\langle\!\langle b \rangle\!\rangle \psi$ of a standpoint formula $\langle\!\langle a \rangle\!\rangle \varphi$ are interpreted from the view of the coalition $\{a,b\}$, i.e., that a can access the atomic propositions in $P_a \cup P_b$ to determine what agent b knows from the history. This models scenarios in which information is actively exchanged between agents. For example, in negotiations between two countries, where each country has access only to a part of the intelligence information, incremental semantics can help model the exchange of information and the gradual improvement of understanding of the situation, while decremental semantics can model how each country draws conclusions based solely on common observations. So, as potential applications, we can highlight the modeling of conflicts and negotiations.

– The *decremental semantics* $\models_{\text{decr}}$ is a variant of $\models_{\text{pobs}}$ in which standpoint subformulas $\langle\!\langle b \rangle\!\rangle \psi$ inside a formula $\langle\!\langle a \rangle\!\rangle \varphi$ are interpreted from the perspective of agent a. While a knows the transition system of agent b, it can only access the atomic propositions in $P_a \cap P_b$ to guess what b knows from the history. That is, to make a guess on the current state of b's transition system T_b, agent a may only use the atomic propositions in the intersection rather than the full set P_b. For example, consider autonomous vehicles that share some sensors, but each also has access to sensors unavailable to the others. If the vehicles make decisions solely based on the values of the common sensors, the decremental semantics applies as each agent must reason based on overlapping knowledge only. In contrast, under incremental semantics agents share both common and individual sensor data, so access to the combined observations lets them refine their understanding beyond individual inference.

The decremental and incremental semantics share ideas of distributed knowledge and the "everybody knows" operator of epistemic logics (Fagin et al. 2004).

Main Contributions. Besides introducing the four new semantics for SLTL (Section 3), our main contributions are
– a generic model-checking algorithm that is applicable for all five semantics (Section 4)
– complexity-theoretic results for the model checking problem of SLTL under the different semantics (Section 5). More precisely we show PSPACE-completeness for full SLTL under $\models_{\text{step}}$ and $\models_{\text{public}}$, and for SLTL formulas of alternation depth 1 under $\models_{\text{pobs}}$, $\models_{\text{decr}}$ and $\models_{\text{incr}}$. This stands in contrast to the EXPSPACE-completeness of the satisfiability problem for SLTL under the step semantics (Demri and Walega 2024). Furthermore, our results yield an EXPTIME upper bound for $\models_{\text{decr}}$. The same holds for $\models_{\text{pobs}}$ under the additional assumption that the P_a's are pairwise disjoint. We show that SLTL under all five semantics can be embedded into LTLK. For the case of $\models_{\text{incr}}$, the embedding yields an $(N-1)$-EXPSPACE upper bound where $N = |Ag|$. For the case of $\models_{\text{pobs}}$ and the SLTL fragment of alternation depth at most d, the embedding into LTLK implies $(d-1)$-EXPSPACE membership.

While our algorithm builds on ideas from the LTLK model-checking algorithm in (Bozzelli, Maubert, and Murano 2024) (and even for CTL*K), it exploits the simpler nature of SLTL compared to LTLK and generates smaller history-automata than those that would have been constructed when applying iteratively the powerset constructions of (Bozzelli, Maubert, and Murano 2024). As such, our algorithm can be seen as an adaption of (Bozzelli, Maubert, and Murano 2024) that takes a more fine-grained approach for the different SLTL semantics resulting in the different complexity bounds described above.

Omitted proofs can be found in the full version (Aghamov et al. 2025).

2 Preliminaries

Throughout the paper, we assume some familiarity with linear temporal logic interpreted over transition systems and automata-based model checking, see e.g. (Clarke et al. 2018; Baier and Katoen 2008).

Notations for Strings. Given an alphabet Σ, we write Σ^* for the set of finite strings over Σ, Σ^ω for the set of infinite strings over Σ and Σ^∞ for $\Sigma^* \cup \Sigma^\omega$. As usual, $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$ where ε denotes the empty string. Given a (in)finite string $\varsigma = H_0 H_1 \ldots H_n$ or $\varsigma = H_0 H_1 \ldots$ over Σ, let $\mathit{first}(\varsigma) = H_0$. If

$\varsigma = H_0 \dots H_n$ is finite then $last(\varsigma) = H_n$. For $i, j \in \mathbb{N}$, $\varsigma[i \dots j]$ denotes the substring $H_i \dots H_j$ if $i \leqslant j$ (and assuming $j \leqslant n$ if ς is a finite string of length n) and $\varsigma[i \dots j] = \varepsilon$ if $i > j$. If $i = j$ then $\varsigma[i \dots i] = \varsigma[i] = H_i$. So, $\varsigma[0 \dots j]$ denotes the prefix $H_0 \dots H_j$. If ς is infinite then $\varsigma[j \dots \infty] = H_j H_{j+1} H_{j+2} \dots$.

If $\Sigma = 2^P$ is the powerset of P and $R \subseteq P$ then the projection function $|_R : (2^P)^\infty \to (2^R)^\infty$ is obtained by applying the projection $2^P \to 2^R$, $H \mapsto H \cap R$, elementwise, i.e., if $\varsigma = H_0 H_1 H_2 \dots$ then $\varsigma|_R = (H_0 \cap R)(H_1 \cap R)(H_2 \cap R) \dots$.

Transition Systems. A transition system is a tuple $\mathscr{T} = (S, \to, Init, R, L)$ where S is a finite state space, $\to \subseteq S \times S$ a total transition relation (where totality means that every state s has at least one outgoing transition $s \to s'$), $Init \subseteq S$ the set of initial states, R a finite set of atomic propositions and $L : S \to 2^R$ the labeling function. If $Init$ is a singleton, say $Init = \{init\}$, we simply write $\mathscr{T} = (S, \to, init, R, L)$.

A path in $\mathscr{T}$ is a (in)finite string $\pi = s_0 s_1 \dots s_n \in S^+$ or $\pi = s_0 s_1 s_2 \dots \in S^\omega$ such that $s_i \to s_{i+1}$ for all i. π is initial if $first(\pi) \in Init$. The trace of π is $trace(\pi) = L(s_0) L(s_1) \dots \in (2^R)^+ \cup (2^R)^\omega$. If $s \in S$ then $Paths(\mathscr{T}, s)$ contains all infinite paths in $\mathscr{T}$ starting in s and $Traces(\mathscr{T}, s) = \{trace(\pi) : \pi \in Paths(\mathscr{T}, s)\}$. If P is a superset of R then

$$Traces^P(\mathscr{T}, s) = \left\{\rho \in \left(2^P\right)^\omega : \rho|_R \in Traces(\mathscr{T}, s)\right\}.$$

Thus, $Traces(\mathscr{T}, s) \subseteq (2^R)^\omega$, while $Traces^P(\mathscr{T}, s) \subseteq (2^P)^\omega$. Moreover, $Paths(\mathscr{T}) = \bigcup_{s \in Init} Paths(\mathscr{T}, s)$. $Traces(\mathscr{T})$ and $Traces^P(\mathscr{T})$ have the analogous meaning. If $h \in (2^P)^+$ then $Reach(\mathscr{T}, h)$ denotes the set of states s in $\mathscr{T}$ that are reachable from $Init$ via a path π with $trace(\pi) = h|_R$.

Linear Temporal Logic (LTL). The syntax of LTL over P is given by (where $p \in P$):

$$\varphi \quad ::= \quad true \;\big|\; p \;\big|\; \neg\varphi \;\big|\; \varphi_1 \wedge \varphi_2 \;\big|\; \bigcirc \varphi \;\big|\; \varphi_1 \,\mathrm{U}\, \varphi_2$$

Other Boolean connectives are derived as usual, e.g., $\varphi_1 \vee \varphi_2 = \neg(\neg\varphi_1 \wedge \neg\varphi_2)$. The modalities $\Diamond$ (eventually) and $\Box$ (always) are defined by $\Diamond\varphi = true\,\mathrm{U}\,\varphi$ and $\Box\varphi = \neg\Diamond\neg\varphi$. The standard LTL semantics is formalized by a satisfaction relation $\models_{\text{LTL}}$ where formulas are interpreted over infinite traces (i.e., elements of $(2^P)^\omega$), see e.g. (Clarke et al. 2018; Baier and Katoen 2008). An equivalent semantics can be provided using a satisfaction relation $\models$ that interprets formulas over trace-position pairs $(\rho, n) \in (2^P)^\omega \times \mathbb{N}$ such that $(\rho, n) \models \varphi$ iff $\rho[n \dots \infty] \models_{\text{LTL}} \varphi$.

We use here an equivalent formalisation of the semantics of LTL (and later its extension SLTL) that interprets formulas over *future-history pairs* $(f, h) \in (2^P)^\omega \times (2^P)^+$ with $last(h) = first(f)$, see the upper part of Figure 1 where $f[1 \dots 0] = \varepsilon$. Then, $f \models_{\text{LTL}} \varphi$ iff $(f, first(f)) \models \varphi$. For interpreting LTL formulas, the history is irrelevant: $f \models_{\text{LTL}} \varphi$ iff $(f, first(f)) \models \varphi$ iff $(f, h) \models \varphi$ for some h with $last(h) = first(f)$ iff $(f, h) \models \varphi$ for all h with $last(h) = first(f)$.

If $\mathscr{T} = (S, \to, Init, R, L)$ is a transition system with $R \subseteq P$ and φ an LTL formula over P then $\mathscr{T} \models_{\text{LTL}} \varphi$ iff $f \models_{\text{LTL}} \varphi$ for each $f \in Traces^P(\mathscr{T})$. $Sat_{\mathscr{T}}(\exists\varphi)$ denotes the set of states $s \in S$ where $\{f \in Traces^P(\mathscr{T}, s) : f \models_{\text{LTL}} \varphi\} \neq \varnothing$.

3 SLTL: LTL with Standpoint Modalities

Standpoint LTL (SLTL) extends LTL by standpoint modalities $\langle\!\langle a \rangle\!\rangle \varphi$ where a is an agent and φ a formula.

3.1 Syntax

Given a finite set P of atomic propositions and a finite set Ag of agents, say $Ag = \{a, b, \dots\}$, the syntax of SLTL formulas over P and Ag for $p \in P$ and $a \in Ag$ is given by

$$\varphi \quad ::= \quad true \;\big|\; p \;\big|\; \neg\varphi \;\big|\; \varphi_1 \wedge \varphi_2 \;\big|\; \bigcirc \varphi \;\big|\; \varphi_1 \,\mathrm{U}\, \varphi_2 \;\big|\; \langle\!\langle a \rangle\!\rangle \varphi$$

The intuitive meaning of $\langle\!\langle a \rangle\!\rangle \varphi$ is that from agent a's standpoint it is conceivable that φ will hold, in the sense that there are indications from a's view that there is a path starting in the current state that fulfills φ. The dual standpoint modality is defined by $[\![a]\!]\varphi = \neg\langle\!\langle a \rangle\!\rangle\neg\varphi$ and has the intuitive meaning that from the standpoint of a, φ is unequivocal. That is, under a's view all paths starting in the current state fulfill φ.

Formulas of the shape $\langle\!\langle a \rangle\!\rangle \varphi$ are called *standpoint formulas*. If φ is a SLTL formula, then *maximal standpoint subformulas* of φ are subformulas that have the form $\langle\!\langle a \rangle\!\rangle \phi$ and that are not in the scope of another standpoint operator. For example, $\varphi = (p \wedge \bigcirc\chi_1) \vee \psi$ with $\chi_1 = \langle\!\langle a \rangle\!\rangle(\langle\!\langle b \rangle\!\rangle q \,\mathrm{U}\, r)$ and $\psi = [\![c]\!] \bigcirc (r \wedge \langle\!\langle a \rangle\!\rangle p \,\mathrm{U}\, \langle\!\langle a \rangle\!\rangle q)$ has two maximal standpoint subformulas, namely χ_1 and $\chi_2 = \langle\!\langle c \rangle\!\rangle\neg\bigcirc(r \wedge \langle\!\langle a \rangle\!\rangle p \,\mathrm{U}\, \langle\!\langle a \rangle\!\rangle q)$.

The *alternation depth* $ad(\varphi)$ of φ is the maximal number of alternations between standpoint modalities for different agents. E.g., $ad(\langle\!\langle a \rangle\!\rangle(p \wedge \bigcirc[\![b]\!]q)) = 2$, while $ad(\langle\!\langle a \rangle\!\rangle(p \wedge \bigcirc[\![a]\!]q)) = ad(\langle\!\langle a \rangle\!\rangle p \wedge \bigcirc[\![b]\!]q) = 1$ if $a \neq b$. Formally:

Definition 3.1. *The alternation depth of* SLTL *formulas is defined inductively:*

- $ad(true) = ad(p) = 0$ *for* $p \in P$,
- $ad(\varphi_1 \wedge \varphi_2) = ad(\varphi_1 \,\mathrm{U}\, \varphi_2) = \max\{ad(\varphi_1), ad(\varphi_2)\}$,
- $ad(\neg\varphi) = ad(\bigcirc\varphi) = ad(\varphi)$.

For standpoint formulas $\phi = \langle\!\langle a \rangle\!\rangle \varphi$, *the definition of* $ad(\phi)$ *is as follows. If* φ *is an LTL formula then* $ad(\langle\!\langle a \rangle\!\rangle \varphi) = 1$. *Otherwise let* $\chi_1 = \langle\!\langle b_1 \rangle\!\rangle \psi, \dots, \chi_k = \langle\!\langle b_k \rangle\!\rangle \psi_k$ *be the maximal standpoint subformulas of* φ. *We may suppose an enumeration such that* $a = b_1 = \dots = b_\ell$ *and* $a \notin \{b_{\ell+1}, \dots, b_k\}$. *Then,* $ad(\phi)$ *is the maximum of* $\max\{ad(\chi_i) : i = 1, \dots, \ell\}$ *and* $\max\{ad(\chi_i) : i = \ell{+}1, \dots, k\} + 1$.

Each SLTL formula over a singleton agent set $Ag = \{a\}$ has alternation depth at most 1. For $d \in \mathbb{N}$, let SLTL_d denote the sublogic of SLTL where all formulas φ satisfy $ad(\varphi) \leq d$. In particular, SLTL_0 is LTL.

Remark 3.2. The original papers (Gigante, Gómez Alvarez, and Lyon 2023; Demri and Walega 2024) on standpoint LTL have an additional type of formula $a \preceq b$ where $a, b \in Ag$. The intuitive semantics of $a \preceq b$ is that the standpoint of a is sharper than that of b. In our setting, where we are given transition systems for a and b, formulas $a \preceq b$ are either true or false depending on whether the set of traces of the transition system representing a's view is contained in the set of b's traces or not. As the trace inclusion problem is PSPACE-complete (Kanellakis and Smolka 1990) like the LTL model-checking problem (Sistla and Clarke 1985), the complexity results of Section 5 are not affected when adding sharpening statements $a \preceq b$ to SLTL.

$(f,h) \models p$	iff	$p \in f[0]$	$(f,h) \models \neg\varphi$	iff	$(f,h) \not\models \varphi$
$(f,h) \models \varphi_1 \wedge \varphi_2$	iff	$(f,h) \models \varphi_1$ and $(f,h) \models \varphi_2$	$(f,h) \models \bigcirc\varphi$	iff	$(f[1\ldots\infty], hf[1]) \models \varphi$
$(f,h) \models \varphi_1 \mathrm{U} \varphi_2$	iff	there exists $\ell \in \mathbb{N}$ such that $(f[\ell\ldots\infty], hf[1\ldots\ell]) \models \varphi_2$ and $(f[j\ldots\infty], hf[1\ldots j]) \models \varphi_1$ for all $j<\ell$			
$(f,h) \models \langle\!\langle a \rangle\!\rangle\varphi$	iff	there exists $h' \in (2^P)^+$, $t \in Reach(\mathscr{T}_a, h')$ and $f' \in Traces^P(\mathscr{T}_a, t)$ such that $last(h') = first(f')$, $obs_a(h) = obs_a(h')$ and $(f',h') \models \varphi$			

Figure 1: Satisfaction relation $\models$ for SLTL over future-history pairs $(f,h) \in (2^P)^\omega \times (2^P)^+$ with $last(h) = first(f)$.

3.2 Semantics of SLTL

We consider different semantics of the standpoint modality that differ in what the agents can observe from the history.

SLTL Structures. SLTL structures are tuples $\mathfrak{T} = (\mathscr{T}_0, (\mathscr{T}_a)_{a\in Ag})$ with $\mathscr{T}_0 = (S_0, \to_0, init_0, P_0, L_0)$ a transition system over the full set $P_0 = P$ of atomic propositions, and $\mathscr{T}_a = (S_a, \to_a, init_a, P_a, L_a)$ transition systems for the agents $a \in Ag$ over some $P_a \subseteq P$. For simplicity, we assume these transition systems to have unique initial states.

Semantics of the Standpoint Modalities. We consider five different semantics for $\langle\!\langle a \rangle\!\rangle\varphi$, using satisfaction relations $\models_{\text{step}}$ (step semantics as in (Gigante, Gómez Alvarez, and Lyon 2023; Demri and Walega 2024)), $\models_{\text{pobs}}$ (pure observation-based semantics which is essentially the perfect-recall partial information semantics for LTLK (Bozzelli, Maubert, and Murano 2024) adapted for SLTL structures), $\models_{\text{public}}$ (public-history semantics which are like a perfect-recall semantics where agents have full information about the history), $\models^Q_{\text{decr}}$ and $\models^Q_{\text{incr}}$ (variants of $\models_{\text{pobs}}$ with decremental resp. incremental knowledge) where $Q \subseteq P$.

We deal here with an interpretation over future-history pairs (see Section 2). Let $\models$ be one of the five satisfaction relations. The semantics of SLTL can be presented in a uniform manner as shown in Figure 1. For the dual standpoint operator we obtain: $(f,h) \models [\![a]\!]\varphi$ iff $(f',h') \models \varphi$ for all $h' \in (2^P)^+$, $t \in Reach(\mathscr{T}_a, h')$ and $f' \in Traces^P(\mathscr{T}_a, t)$ with $last(h') = first(f')$ and $obs_a(h) = obs_a(h')$.

The five semantics rely on different observation functions obs_a. In all cases, obs_a is a projection $obs_a : (2^P)^+ \to (2^{\mathfrak{O}_a})^+$, $obs_a(h) = h|_{\mathfrak{O}_a}$ for some $\mathfrak{O}_a \subseteq P$. Intuitively, $\mathfrak{O}_a$ formalizes which of the propositions are visible to agent a in the history. (For the decremental and incremental semantics, both $\mathfrak{O}_a$ and the induced observation function obs_a do not only depend on a, but on the context of the formula $\langle\!\langle a \rangle\!\rangle\varphi$ as will be explained later.) Before presenting the specific choices of $\mathfrak{O}_a$ in the five SLTL semantics, we make some general observations:

Lemma 3.3. *If $f_1, f_2 \in (2^P)^\omega$ and $h \in (2^P)^+$ with $last(h) = first(f_1) = first(f_2)$ then $(f_1,h) \models \langle\!\langle a \rangle\!\rangle\varphi$ iff $(f_2,h) \models \langle\!\langle a \rangle\!\rangle\varphi$.*

By Lemma 3.3 one can drop the f-component and write $(*,h) \models \langle\!\langle a \rangle\!\rangle\varphi$ when $(f,h) \models \langle\!\langle a \rangle\!\rangle\varphi$ for some (each) future f with $first(f) = last(h)$. Truth values of standpoint formulas $\langle\!\langle a \rangle\!\rangle\varphi$ only depend on agent a's observation of the history:

Lemma 3.4. *If $h_1, h_2 \in (2^P)^+$ with $obs_a(h_1) = obs_a(h_2)$ then $(*,h_1) \models \langle\!\langle a \rangle\!\rangle\varphi$ iff $(*,h_2) \models \langle\!\langle a \rangle\!\rangle\varphi$.*

Remark 3.5. SLTL structures defined in (Gigante, Gómez Alvarez, and Lyon 2023; Demri and Walega 2024) assign to each agent a a nonempty subset $\lambda(a) \subseteq (2^P)^\omega$ and assume a universal agent, called $*$, such that $\lambda(a) \subseteq \lambda(*)$ for all other agents a. The latter is irrelevant for our purposes. Assuming transition system representations for the sets $\lambda(a)$ is natural for the model checking problem. In contrast to (Gigante, Gómez Alvarez, and Lyon 2023; Demri and Walega 2024), we suppose here that the standpoint transition systems $\mathscr{T}_a$ are defined over some $P_a \subseteq P$, which appears more natural for defining the information that an agent can extract from the history. In the semantics of $\langle\!\langle a \rangle\!\rangle\varphi$, we thus switch from $Traces(\mathscr{T}_a, t) \subseteq (2^{P_a})^\omega$ to $Traces^P(\mathscr{T}_a, t) \subseteq (2^P)^\omega$ which essentially means that a may guess the truth values of the atomic propositions in $P \setminus P_a$ to predict whether φ can hold in the future. Alternatively, one could define SLTL structures as tuples $(\mathscr{T}_0, (\mathscr{T}_a, P_a)_{a\in Ag})$ where $\mathscr{T}_0$ is as before, the $\mathscr{T}_a$'s are transition systems over some $R_a \subseteq P$, and $P_a \subseteq R_a$ where the P_a serves to define the functions obs_a for the histories. With $R_a = P$ and $\lambda(a) = Traces(\mathscr{T}_a)$, SLTL under $\models_{\text{step}}$ agrees with the logic considered in (Gigante, Gómez Alvarez, and Lyon 2023; Demri and Walega 2024) (except for sharpening statements; see Remark 3.2). Our model checking algorithm can easily be adapted for this more general type of SLTL structures without affecting our complexity results.

Step Semantics: The semantics of the standpoint modality $\langle\!\langle a \rangle\!\rangle\varphi$ introduced in (Gigante, Gómez Alvarez, and Lyon 2023; Demri and Walega 2024) relies on the assumption that the only information that the agents can extract from the history is the number of steps that have been performed in the past. They formulate the semantics in terms of trace-position pairs $(\rho, n) \in (2^P)^\omega \times \mathbb{N}$ and define $(\rho,n) \models \langle\!\langle a \rangle\!\rangle\varphi$ iff $(\rho', n) \models \varphi$ for some $\rho' \in Traces(\mathscr{T}_a)$. Reformulated to our setting, the set of propositions visible for agent a in the history is $\mathfrak{O}_a = \varnothing$, which yields the observation function $obs_a : (2^P)^+ \to \mathbb{N}$, $obs_a(h) = |h|$. Formulated for future-history pairs, $(f,h) \models_{\text{step}} \langle\!\langle a \rangle\!\rangle\varphi$ iff there is a word $h' \in (2^P)^+$, a state $t \in Reach(\mathscr{T}_a, h')$ and a trace $f' \in Traces^P(\mathscr{T}_a, t)$ such that $last(h') = first(f')$, $|h| = |h'|$ and $(f',h') \models_{\text{step}} \varphi$.

Pure Observation-Based Semantics: In the style of the (dual of the) classical K-modality in LTLK (Halpern and Vardi 1986; Halpern and Vardi 1989; Bozzelli, Maubert, and Murano 2024) (cf. Section 3.3) we can deal with the observation function that projects the given history h to the observations that agent a can make when exactly the propositions in P_a are visible for a. That is, for the pure observation-

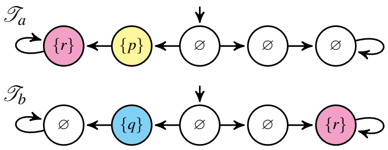

Figure 2: The transition systems used in Example 3.6.

based semantics, $\mathfrak{O}_a = P_a$. Then, $(f,h) \models_{\text{pobs}} \langle\!\langle a \rangle\!\rangle \varphi$ iff there exist a word $h' \in (2^P)^+$, a state $t \in Reach(\mathcal{T}_a, h')$ and a trace $f' \in Traces^P(\mathcal{T}_a, t)$ such that $last(h') = first(f')$, $h|_{P_a} = h'|_{P_a}$ and $(f', h') \models_{\text{pobs}} \varphi$.

Public-History Semantics: In the public-history semantics all agents a have full information about the history h, i.e., $\mathfrak{O}_a = P$ and $obs_a(h) = h$. Then, $(f,h) \models_{\text{public}} \langle\!\langle a \rangle\!\rangle \varphi$ iff $\exists t \in Reach(\mathcal{T}_a, h)$ and a trace $f' \in Traces^P(\mathcal{T}_a, t)$ such that $last(h) = first(f')$ and $(f', h) \models_{\text{public}} \varphi$.

Decremental Semantics: The decremental semantics is a variant of the pure observation-based semantics with a different meaning of nested standpoint subformulas: A standpoint formula $\langle\!\langle a \rangle\!\rangle \varphi$ interpretes maximal standpoint subformulas $\langle\!\langle b \rangle\!\rangle \psi$ of φ from the view of agent a. The assumption is that a knows agent b's transition system $\mathcal{T}_b$ and thus can make the same guesses for the future as b, but can access only the truth values of joint atomic propositions in $P'_a \cap P_b$ from the history to make a guess what b has observed in the past. Here, $P'_a \subseteq P_a$ is the set of atomic propositions accessible when interpreting $\langle\!\langle a \rangle\!\rangle \varphi$ which can itself be a subformula of a larger standpoint formula $\langle\!\langle c \rangle\!\rangle \chi$ (in which case $\langle\!\langle a \rangle\!\rangle \varphi$ is interpreted from the view of agent c and $P'_a \subseteq P_c \cap P_a$). Formally, we use a parametric satisfaction relation $\models^Q_{\text{decr}}$ for $Q \subseteq P$. The intuitive meaning of $\langle\!\langle a \rangle\!\rangle \varphi$ under $\models^Q_{\text{decr}}$ is that a can extract from the history h exactly the truth values of the propositions in $\mathfrak{O}^Q_a = Q \cap P_a$. So, $(f,h) \models^Q_{\text{decr}} \langle\!\langle a \rangle\!\rangle \varphi$ iff there exist a word $h' \in (2^P)^+$, a state $t \in Reach(\mathcal{T}_a, h')$ and a trace $f' \in Traces^P(\mathcal{T}_a, t)$ such that $last(h') = first(f')$, $h|_{Q \cap P_a} = h'|_{Q \cap P_a}$ and $(f', h') \models^{Q \cap P_a}_{\text{decr}} \varphi$. For the satisfaction over a SLTL structure, we start with $Q = P$.

Incremental Semantics: The incremental semantics also relies on a parametric satisfaction relation $\models^Q_{\text{incr}}$ where $Q \subseteq P$. The intuitive meaning of $\langle\!\langle a \rangle\!\rangle \varphi$ under $\models^Q_{\text{incr}}$ is that agent a can extract from the history h exactly the propositions in $\mathfrak{O}^Q_a = Q \cup P_a$ and interpretes φ over $Q \cup P_a$. Thus, when interpreting nested standpoint subformulas $\langle\!\langle b \rangle\!\rangle \psi$ of $\langle\!\langle a \rangle\!\rangle \varphi$ then agents a and b build a coalition to extract the information from the history. Formally, $(f,h) \models^Q_{\text{incr}} \langle\!\langle a \rangle\!\rangle \varphi$ iff there exist a word $h' \in (2^P)^+$, a state $t \in Reach(\mathcal{T}_a, h')$ and a trace $f' \in Traces^P(\mathcal{T}_a, t)$ such that $last(h') = first(f')$, $h|_{Q \cup P_a} = h'|_{Q \cup P_a}$ and $(f', h') \models^{Q \cup P_a}_{\text{incr}} \varphi$. For the satisfaction over a SLTL structure, we start with $Q = \varnothing$.

Example 3.6. We illustrate the differences between the semantics: Consider two agents a and b with $P_a = \{p, r\}$, $P_b = \{q, r\}$, the transition system of Figure 2, and formulas $\phi_u = \langle\!\langle a \rangle\!\rangle \bigcirc \langle\!\langle b \rangle\!\rangle u$, with $u \in P = P_a \cup P_b$. As these formulas ϕ_u are negation-free and contain only the existential $\langle\!\langle\rangle\!\rangle$ modality, satisfaction becomes easier if less restrictive requirements are imposed on histories. So, for all future-history pairs (f,h), $u \in \{p,q,r\}$ and using the total order public $\succ$ incr $\succ$ pobs $\succ$ decr $\succ$ step:

$$\text{If } (f,h) \models_\sigma \phi_u \text{ and } \sigma \succ \tau \text{ then } (f,h) \models_\tau \phi_u$$

where $\sigma, \tau \in \{\text{public}, \text{incr}, \text{pobs}, \text{decr}, \text{step}\}$. Now, we provide examples where the reverse implications do not hold. We regard the future-history pair (f,h) where $h = \varnothing\{p,q\}$ and $f = \{p,q\}\{r\}^\omega$.

For $\phi = \langle\!\langle a \rangle\!\rangle \bigcirc \langle\!\langle b \rangle\!\rangle r$, we have $(f,h) \not\models_{\text{public}} \phi$: In the quantification of $\langle\!\langle a \rangle\!\rangle$, all possible future traces f' have to start with $\{p,q\}$. So, according to $\mathcal{T}_a$, the future has to contain r in the second position. When evaluating $\langle\!\langle b \rangle\!\rangle$, these first two positions of f' have to be respected, i.e., the history under consideration is $h' = \varnothing\{p,q\}H$ where H contains r. But there is no trace in $\mathcal{T}_b$ with q in the second and r in the third position. However, $(f,h) \models_{\text{incr}} \phi$ as we can choose any trace $\{p\}\{r\}...$ as the future trace in the quantification of $\langle\!\langle a \rangle\!\rangle$. This means, the history $\varnothing\{p\}\{r\}$ has to be respected when evaluating $\langle\!\langle b \rangle\!\rangle$. When projecting to P_b, this can be respected by the trace in $\mathcal{T}_b$ satisfying r in the third position.

Next, consider $\phi = \langle\!\langle a \rangle\!\rangle \bigcirc \langle\!\langle b \rangle\!\rangle p$. Then, $(f,h) \not\models_{\text{incr}} \phi$ as the future trace f' quantified in $\langle\!\langle a \rangle\!\rangle$ starting after one step in the history cannot have p in the second position because there is no trace with p in the third position in $\mathcal{T}_a$. In the incremental semantics, the history chosen for $\langle\!\langle b \rangle\!\rangle$ has to agree with f' on p in the second step and hence can also not contain p there. In contrast, $(f,h) \models_{\text{pobs}} \phi$ as here agent b can "choose" the truth value of p arbitrarily as $p \notin P_b$.

With similar reasoning, $\models_{\text{pobs}}$ and $\models_{\text{decr}}$ can be distinguished by the formula $\phi = \langle\!\langle a \rangle\!\rangle \bigcirc \langle\!\langle b \rangle\!\rangle (q \wedge r)$ and (f,h) as above: $(f,h) \not\models_{\text{pobs}} \phi$ as q is false on any state reachable in $\mathcal{T}_b$ after two steps. So, no matter which history h' and future f' is chosen in the quantification of $\langle\!\langle a \rangle\!\rangle$, there is no future-history pair (f'', h'') that agrees with a trace in $\mathcal{T}_b$ on all atomic propositions in P_b and is such that q holds at the third position. In contrast, $(f,h) \models_{\text{decr}} \phi$: We can choose $h' = \varnothing\{p\}$ and $f' = \{p\}\{r\}^\omega$ for the quantification of $\langle\!\langle a \rangle\!\rangle$ as h' agrees with $h = \varnothing\{p,q\}$ on all atomic propositions in P_a. Then, the quantification of $\langle\!\langle b \rangle\!\rangle$ in ϕ can choose $h'' = \varnothing\varnothing\{r,q\}^\omega$ and $f'' = \{r,q\}^\omega$ as this agrees with the trace of the right-hand side path in $\mathcal{T}_b$ for all atomic propositions in $P_a \cap P_b = \{r\}$.

Finally, for $\models_{\text{decr}}$ and $\models_{\text{step}}$ consider $\phi = \langle\!\langle a \rangle\!\rangle \bigcirc \langle\!\langle b \rangle\!\rangle \neg r$ and (f,h) as above. Under $\models_{\text{decr}}$, in the first quantification of $\langle\!\langle a \rangle\!\rangle$, the history h' must contain p and potentially q. Hence, the future f' must contain r in the second position as only the left path in $\mathcal{T}_a$ is possible. So, the history h'' quantified in $\langle\!\langle b \rangle\!\rangle$ also must contain r in the third position as $r \in P_a \cap P_b$. Thus, ϕ cannot hold, i.e., $(f,h) \not\models_{\text{decr}} \phi$. In the step semantics, the quantification for $\langle\!\langle b \rangle\!\rangle$ can simply choose history $h'' = \varnothing\{q\}\varnothing$ and future $f'' = \varnothing^\omega$ and so $(f,h) \models_{\text{step}} \phi$.

Satisfaction of SLTL Formulas Over Structures. Given a structure $\mathfrak{T} = \big(\mathcal{T}_0, (\mathcal{T}_a)_{a \in Ag}\big)$ as in Section 3.2, an SLTL formula ϕ and $\models \in \{\models_{\text{step}}, \models_{\text{pobs}}, \models_{\text{public}}, \models_{\text{decr}}, \models_{\text{incr}}\}$:

$$\mathfrak{T} \models \phi \quad \text{iff} \quad (f, \mathit{first}(f)) \models_* \phi \text{ for all } f \in \mathit{Traces}(\mathscr{T}_0)$$

where $\models_*$ equals $\models$ for step, pure observation-based and public-history semantics, $\models_*$ equals $\models^P_{\text{decr}}$ for decremental semantics and $\models_*$ equals $\models^{\varnothing}_{\text{incr}}$ for incremental semantics.

Remark 3.7. The meanings of $\langle\!\langle a \rangle\!\rangle \varphi$ under $\models^Q_{\text{decr}}$ and $\models^Q_{\text{incr}}$ are context-dependent through the parameter Q. If, e.g., $\phi = \langle\!\langle a \rangle\!\rangle \varphi \vee \langle\!\langle b \rangle\!\rangle (q \wedge \bigcirc \langle\!\langle a \rangle\!\rangle \varphi)$ for φ an SLTL formula then the first occurrence of $\langle\!\langle a \rangle\!\rangle \varphi$ is interpreted over $\models^P_{\text{decr}}$, while the second is interpreted over $\models^{P_b}_{\text{decr}}$. The set $Q = Q^*_\chi$ with $* \in \{\text{decr}, \text{incr}\}$ over which occurrences of subformulas χ of ϕ are interpreted can be derived from the syntax tree of ϕ.

Formally, we assign a set Q^*_v with $* \in \{\text{decr}, \text{incr}\}$ to each node v in the tree. Let ϕ_v denote the subformula represented by v. If v is the root node then $\phi_v = \phi$ and $Q^{\text{decr}}_v = P$ while $Q^{\text{incr}}_v = \varnothing$. Let now v and w be nodes such that v is the father of w. If v is not labeled by a standpoint modality then $Q^*_v = Q^*_w$. Otherwise ϕ_v has the form $\langle\!\langle a \rangle\!\rangle \phi_w$ for some $a \in Ag$ and $Q^{\text{decr}}_w = Q^{\text{decr}}_v \cap P_a$ while $Q^{\text{incr}}_w = Q^{\text{incr}}_v \cup P_a$. With abuse of notations, we will simply write Q^*_χ instead of Q^*_v for a node v representing the particular occurrence of $\chi = \phi_v$ in ϕ.

Obviously, the five semantics agree on the LTL fragment. More precisely, for $\models$ as above, $\mathfrak{T}$ a SLTL structure and ϕ an LTL formula we have $\mathfrak{T} \models \phi$ iff $\mathscr{T}_0 \models_{\text{LTL}} \phi$ in the sense that $\mathit{Traces}(\mathscr{T}_0) \subseteq \{f \in (2^P)^\omega : f \models_{\text{LTL}} \phi\}$. For SLTL formulas with at least one standpoint subformula, but no alternation of standpoint modalities, the pure observation-based, decremental and incremental semantics agree, but might yield different truth values than the step or public-history semantics:

Lemma 3.8. *If ϕ is a* SLTL *formula of alternation depth* 1 *then $\mathfrak{T} \models_{\text{pobs}} \phi$ iff $\mathfrak{T} \models_{\text{decr}} \phi$ iff $\mathfrak{T} \models_{\text{incr}} \phi$, while $\mathfrak{T} \not\models_{\text{pobs}} \phi$ and $\mathfrak{T} \models \phi$ (or vice versa) is possible where $\models \in \{\models_{\text{step}}, \models_{\text{public}}\}$. Likewise, $\mathfrak{T} \not\models_{\text{public}} \phi$ and $\mathfrak{T} \models_{\text{step}} \phi$ (or vice versa) is possible.*

3.3 From Pure Observation-Based SLTL to LTLK

LTLK (also called CKL_m) (Halpern and Vardi 1986; Halpern and Vardi 1989; Bozzelli, Maubert, and Murano 2024) extends LTL by an unary knowledge modality K_a for every $a \in Ag$ and a common knowledge operator C_A for coalitions $A \subseteq Ag$. We drop the latter and deal with LTLK where the grammar for formulas is the same as for SLTL when $\langle\!\langle a \rangle\!\rangle \varphi$ is replaced with $K_a \varphi$. The alternation depth of LTLK formulas and the sublogics LTLK_d are defined as for SLTL.

LTLK structures are tuples $(\mathscr{T}, (\sim_a)_{a \in Ag})$ where $\mathscr{T} = (S, \rightarrow, \mathit{init}, R, L)$ is a transition system and the $\sim_a$, $a \in Ag$, are equivalence relations on S. The intended meaning is that $s \sim_a t$ if agent a cannot distinguish the states s and t.

The perfect-recall semantics of LTLK extends the standard LTL semantics formulated for path-position pairs (π, n) consisting of a path $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}(\mathscr{T})$ and a position $n \in \mathbb{N}$ by $(\pi, n) \models_{\text{LTLK}} K_a \varphi$ iff $(\pi', n) \models_{\text{LTLK}} \varphi$ for all paths $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}(\mathscr{T})$ with $s_i \sim_a s'_i$ for $i = 0, 1, \ldots, n$. As such, the dual standpoint modality $[\![a]\!]$ under the pure observation-based semantics resembles the K_a modality of LTLK. Indeed, SLTL under $\models_{\text{pobs}}$ can be considered as a special case of LTLK under the perfect-recall semantics:

Lemma 3.9. *Given a pair $(\mathfrak{T}, \phi)$ consisting of a* SLTL *structure $\mathfrak{T} = (\mathscr{T}_0, (\mathscr{T}_a)_{a \in Ag})$ and a* SLTL *formula ϕ, one can construct in polynomial time an LTLK structure $\mathfrak{T}_{\text{LTLK}} = (\mathscr{T}', (\sim_a)_{a \in Ag})$ and an LTLK formula ϕ_{LTLK} such that:*

(1) $\mathfrak{T} \models_{\text{pobs}} \phi$ if and only if $\mathfrak{T}_{\text{LTLK}} \models_{\text{LTLK}} \phi_{\text{LTLK}}$

(2) ϕ and ϕ_{LTLK} have the same alternation depth.

Proof sketch. Assume w.l.o.g. that the initial state labelings in $\mathscr{T}_0$ and $\mathscr{T}_a$ are consistent for all a. First, for each $a \in Ag \cup \{0\}$, we extend $\mathscr{T}_a$ to its completion $\mathscr{T}_a^\perp$ by adding new states $\perp^Q_a$ for each $Q \subseteq P_a$, where the set of atomic propositions in $\mathscr{T}_a^\perp$ is $P^\perp_a = P_a \cup \{\perp_a\}$. The labeling of the original states is unchanged, i.e., $L^\perp_a(s) = L_a(s)$, while for the fresh states $L^\perp_a(\perp^Q_a) = Q \cup \{\perp_a\}$. We then construct the synchronous product $\mathscr{T}' = S^\perp_0 \times \prod_{a \in Ag} \mathscr{T}^\perp_a$ such that $L^\perp_0(s) \cap P_a = L^\perp_a(s_a) \cap P_a$ for each $a \in Ag$. The set of atomic propositions is $P' = P \cup \{\perp_a : a \in Ag \cup \{0\}\}$, and $L'(s, (s_a)_{a \in Ag}) = L_0(s) \cup \{\perp_a : a \in Ag \cup \{0\}, \perp_a \in L^\perp_a(s_a)\}$. The initial state of $\mathscr{T}'$ is $\mathit{init}' = (\mathit{init}_0, (\mathit{init}_a)_{a \in Ag})$, and the transitions are defined synchronously. The equivalence relations $\sim_a$ on S' satisfy $\sigma \sim_a \theta$ iff $L'(\sigma) \cap P^\perp_a = L'(\theta) \cap P^\perp_a$. We then inductively translate SLTL formulas φ into "equivalent" LTLK formulas φ': $\varphi' = \varphi$, $(\neg\varphi)' = \neg\varphi'$, $(\varphi \wedge \psi)' = \varphi' \wedge \psi'$, $(\bigcirc\varphi)' = \bigcirc\varphi'$, $(\varphi_1 \mathbin{\mathrm{U}} \varphi_2)' = \varphi'_1 \mathbin{\mathrm{U}} \varphi'_2$, $(\langle\!\langle a \rangle\!\rangle \varphi)' = \overline{K}_a(\varphi' \wedge \Box\neg\perp_a)$. Finally, we define $\phi_{\text{LTLK}} = \Box\neg\perp_0 \rightarrow \phi'$. □

4 Model Checking SLTL

Let $\models$ be one of the five satisfaction relations defined in Section 3.2. The task of the SLTL model checking problem is as follows. Given a SLTL structure $\mathfrak{T} = \big(\mathscr{T}_0, (\mathscr{T}_a)_{a \in Ag}\big)$ and an SLTL formula ϕ, decide whether $\mathfrak{T} \models \phi$.

We present a generic model checking algorithm for the five semantics, which can be seen as an adaptation of the standard CTL*-like model checking procedure (Emerson and Sistla 1984; Emerson and Lei 1987) and its variant for LTLK and CTL*K under the perfect-recall semantics (Bozzelli, Maubert, and Murano 2024). The central idea is an inductive approach to compute deterministic finite automata (DFA) $\mathscr{D}_\chi$ for all standpoint subformulas $\chi = \langle\!\langle a \rangle\!\rangle \varphi$ of ϕ. The automaton for χ, called history-DFA, represents the language of all histories $h \in (2^P)^+$ with $(*, h) \models_\chi \chi$ where $\models_\chi$ stands for one of the satisfaction relations $\models_{\text{step}}$, $\models_{\text{pobs}}$, $\models_{\text{public}}$, $\models^Q_{\text{decr}}$ or $\models^Q_{\text{incr}}$ with $Q \in \{Q^{\text{decr}}_\chi, Q^{\text{incr}}_\chi\}$ as in Remark 3.7.

The inductive computation corresponds to a bottom-up approach where the innermost standpoint subformulas, i.e., subformulas $\chi = \langle\!\langle a \rangle\!\rangle \psi$ where ψ is an LTL formula, are treated first. When computing $\mathscr{D}_\chi$ for a subformula $\chi = \langle\!\langle a \rangle\!\rangle \psi$ where ψ contains standpoint modalities, we may assume that history-DFA for the maximal standpoint subformulas $\chi_1, \ldots, \chi_k$ of ψ have been computed before.

Preprocessing for $\models_{\text{decr}}$ and $\models_{\text{incr}}$. In the decremental and incremental semantics, our iterative bottom-up approach requires computing the relevant parameters $Q \subseteq P$ of the satisfaction relations $\models^Q_{\text{decr}}$ and $\models^Q_{\text{incr}}$ over which subformulas of ϕ

are interpreted. This can be done in time linear in the length of ϕ by a top-down analysis of the syntax tree of ϕ (see Remark 3.7). In what follows, we shall write Q_χ to denote the corresponding subset Q_χ^{decr} or Q_χ^{incr}, respectively, of P for (an occurrence of) a subformula χ. Thus, if χ is not in the scope of a standpoint modality or a maximal standpoint subformulas of ϕ then $Q_\chi = P$ under the decremental semantics and $Q_\chi = \varnothing$ under the incremental semantics. If $\chi = \langle\!\langle a \rangle\!\rangle \psi$ is (an occurrence) of a standpoint subformula of ϕ and χ' a subformula of ψ that – as a subformula of ψ – is not in the scope of a standpoint modality or a maximal standpoint subformula of ψ then $Q_{\chi'} = Q_\chi \cap P_a$ for the decremental semantics and $Q_{\chi'} = Q_\chi \cup P_a$ for the incremental semantics.

Observation Sets $\mathfrak{O}_\chi$. Our algorithm uses subsets $\mathfrak{O}_\chi \subseteq P$ for the standpoint subformulas $\chi = \langle\!\langle a \rangle\!\rangle \varphi$ of ϕ, whose definition depends on the considered semantics: $\mathfrak{O}_\chi = \varnothing$ for $\models_{\text{step}}$, $\mathfrak{O}_\chi = P_a$ for $\models_{\text{pobs}}$, $\mathfrak{O}_\chi = P$ for $\models_{\text{public}}$, $\mathfrak{O}_\chi = Q_\chi^{\text{decr}} \cap P_a$ for $\models_{\text{decr}}$ and $\mathfrak{O}_\chi = Q_\chi^{\text{incr}} \cup P_a$ for $\models_{\text{incr}}$. Let $obs_\chi : (2^P)^* \to (2^{\mathfrak{O}_\chi})^*$ be the induced observation function $obs_\chi(h) = h|_{\mathfrak{O}_\chi}$.

Transition Systems $\mathcal{T}_a^R$. Define $R_\chi = P_a$ under decremental and step semantics and $R_\chi = P$ otherwise. We denote $R = R_\chi$, where $R \supseteq P_a$, and switch from the transition system $\mathcal{T}_a$ to an extended system $\mathcal{T}_a^R$ over the domain R, which preserves the behavior of $\mathcal{T}_a$. , but makes nondeterministic guesses for the truth values of propositions in $R \setminus P_a$ for the starting state, as well as in every step of a computation. So, the states of $\mathcal{T}_a^R$ are pairs (s, O) with $s \in S_a$ and $O \subseteq R$ such that $L_a(s) = O \cap P_a$.

The switch from $\mathcal{T}_a$ to $\mathcal{T}_a^R$ will be needed to deal with the constraints $last(h') = first(f')$ in the definitions of $\models_{\text{public}}$, $\models_{\text{incr}}$ and $\models_{\text{pobs}}$.

Remark 4.1. If $R = P_a$ then $\mathcal{T}_a^R$ and $\mathcal{T}_a$ are isomorphic, so the switch from $\mathcal{T}_a$ to $\mathcal{T}_a^R$ is obsolete. (Note that then $O = L_a(s)$ for all states (s, O) in $\mathcal{T}_a^R$.) This applies to $\models_{\text{step}}$ where $\mathfrak{O}_\chi = \varnothing$ and $\models_{\text{decr}}$ where $\mathfrak{O}_\chi \subseteq P_a$.

Definition 4.2. *Let χ be a standpoint subformula of ϕ. A history-DFA for χ is a deterministic finite automaton $\mathcal{D} = (\mathcal{X}_\chi, \delta_\chi, init_\chi, F_\chi)$ over the alphabet 2^P such that*

(1) $\mathcal{L}(\mathcal{D}) = \{h \in (2^P)^+ : (*, h) \models_\chi \chi\}$.

(2) $\mathcal{D}$ is $\mathfrak{O}_\chi$-deterministic in the following sense: Whenever $h_1, h_2 \in (2^P)^+$ such that $h_1|_{\mathfrak{O}_\chi} = h_2|_{\mathfrak{O}_\chi}$ then $\delta_\chi(init_\chi, h_1) = \delta_\chi(init_\chi, h_2)$, where the transition function $\delta_\chi : \mathcal{X}_\chi \times 2^P \to \mathcal{X}_\chi$ is extended to a function $\mathcal{X}_\chi \times (2^P)^* \to \mathcal{X}_\chi$ in the standard way and $init_\chi$ its initial state.

For an example construction of history-DFA see Figure 3 and Example 4.4 on page 8.

Generic SLTL Model Checking Algorithm

Basis of Induction. In the basis of induction we are given a standpoint formula $\chi = \langle\!\langle a \rangle\!\rangle \varphi$ with φ an LTL formula. Let $\mathcal{T}_\chi = \mathcal{T}_a^R$ where $R = R_\chi = P_a \cup \mathfrak{O}_\chi$. We apply a mild variant of standard LTL model checking techniques (Vardi and Wolper 1986; Vardi and Wolper 1994; Baier and Katoen 2008; Clarke et al. 2018) to compute the set $Sat_{\mathcal{T}_a^R}(\exists\varphi)$ of states (s, O) in $\mathcal{T}_a^R$ for which there is a $f \in Traces^P(\mathcal{T}_a^R, (s, O))$ with $f \models_{\text{LTL}} \varphi$. We then use a powerset construction applied to $\mathcal{T}_a^R$ (see Definition 4.3 below) to construct an $\mathfrak{O}_\chi$-deterministic history-DFA $\mathcal{D}_\chi$.

Step of Induction. Suppose now $\chi = \langle\!\langle a \rangle\!\rangle \varphi$ and φ has standpoint subformulas. Let $\chi_1, \ldots, \chi_k$ be the maximal standpoint subformulas of φ, say $\chi_i = \langle\!\langle b_i \rangle\!\rangle \psi_i$ for $i = 1, \ldots, k$, and let $\mathcal{D}_1 = \mathcal{D}_{\chi_1}, \ldots, \mathcal{D}_k = \mathcal{D}_{\chi_k}$ be their history-DFAs over 2^P. Let $\mathcal{D}_i = (X_i, \delta_i, init_i, F_i)$ and $\mathfrak{O}_i = \mathfrak{O}_{\chi_i}$. Further, we simply write R for $R_\chi = P_a \cup \mathfrak{O}_\chi$. We consider the transition system

$$\mathcal{T}_\chi = \mathcal{T}_a^{R_\chi} \bowtie \mathcal{D}_1 \bowtie \ldots \bowtie \mathcal{D}_k$$

that is obtained by putting $\mathcal{T}_a^{R_\chi}$ in parallel to the product of the $\mathcal{D}_i$'s. For this, we introduce pairwise distinct, fresh atomic propositions $p_1, \ldots, p_k$ for each of the χ_i's and define the components $\mathcal{T}_\chi = (Z_\chi, \to_\chi, Init_\chi, \mathcal{P}_\chi, L_\chi)$ as follows. The state space is $Z_\chi = S_a^R \times X_1 \times \ldots \times X_k$. The transition relation satisfies: $((s, O), x_1, \ldots, x_k) \to_\chi ((s', O'), x_1', \ldots, x_k')$ iff there is $H \subseteq P$ such that $H \cap R = O'$, $(s, O) \to_a^R (s', O')$ and $x_i' = \delta_i(x_i, H)$ for $i = 1, \ldots, k$. The set of atomic propositions is $\mathcal{P}_\chi = R \cup \{p_1, \ldots, p_k\}$.

The labeling function $L_\chi : Z_\chi \to 2^{\mathcal{P}_\chi}$ is given by $L_\chi((s, O), x_1, \ldots, x_k) \cap R = O$ and $p_i \in L_\chi(s, x_1, \ldots, x_k)$ iff $x_i \in F_i$ for $i = 1, \ldots, k$. Lastly, $Init_\chi$ contains all states $((init_a, O), x_1, \ldots, x_k)$ such that $(init_a, O) \in Init_a^R$ (i.e., $O \cap P_a = L_a(init_a)$) and there exists $H \subseteq P$ with $x_i = \delta_i(init_i, H)$ for $i = 1, \ldots, k$ and $O = H \cap R$.

We now replace φ with the LTL formula $\varphi' = \varphi[\chi_1/p_1, \ldots, \chi_k/p_k]$ over $\mathcal{P} = P \cup \{p_1, \ldots, p_k\}$ that results from φ by syntactically replacing the maximal standpoint subformulas χ_i of φ with p_i. We then apply standard LTL model checking techniques to compute $Sat_{\mathcal{T}_\chi}(\exists\varphi') = \{z \in Z_\chi : \exists f \in Traces^{\mathcal{P}}(\mathcal{T}_\chi, z) \text{ s.t. } f \models_{\text{LTL}} \varphi'\}$.

Having computed $Sat_{\mathcal{T}_\chi}(\exists\varphi')$, the atomic propositions $p_1, \ldots, p_k$ are no longer needed. We therefore switch from $\mathcal{T}_\chi$ to $\mathcal{T}_\chi' = \mathcal{T}_\chi|_R$ which agrees with $\mathcal{T}_\chi$ except that $\mathcal{P}_\chi' = R$ and $L_\chi'(s, x_1, \ldots, x_k) = L_\chi(s, x_1, \ldots, x_k) \cap R$. By applying a powerset construction to $\mathcal{T}_\chi'$ we get an $\mathfrak{O}_\chi$-deterministic history DFA $\mathcal{D}_\chi$ for the language $\{h \in (2^P)^+ : (*, h) \models_\chi \chi\}$ (see Definition 4.3 below, applied to $\mathcal{T} = \mathcal{T}_\chi'$, $\mathfrak{O}_\chi$ and the set $U_\chi = Sat_{\mathcal{T}_\chi}(\exists\varphi')$ to specify the final states in $\mathcal{D}_\chi$).

Final Step. After treating all maximal standpoint subformulas of ϕ, we need to check whether ϕ holds for all traces of $\mathcal{T}_0$. This is done like in the first part of the step of induction, i.e., we build the product $\mathcal{T}_\phi$ of $\mathcal{T}_0$ with the history-DFAs that have been constructed for the maximal standpoint subformulas of ϕ and replace them with fresh atomic propositions, yielding an LTL formula ϕ' over an extension of P. We then apply standard LTL model checking techniques to check whether all traces of $\mathcal{T}_\phi$ satisfy ϕ'. This can be done using an algorithm that is polynomially space-bounded in the size of $\mathcal{T}_\phi$ and the length of ϕ' (and ϕ).

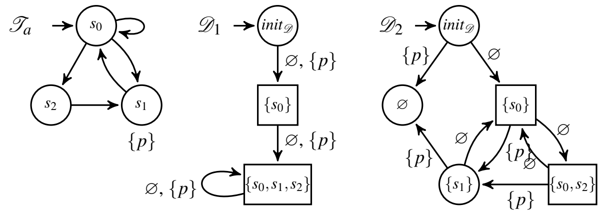

Figure 3: Transition system and the reachable fragments of the default history-DFAs for $\chi = \langle\!\langle a \rangle\!\rangle \bigcirc p$ under the step and the public-history semantics constructed in Example 4.4. Boxes are used for final states, circles for non-final states.

Computation of History-DFA The history-DFA $\mathscr{D}_\chi$ for standpoint subformulas $\chi = \langle\!\langle a \rangle\!\rangle \varphi$ can be obtained by applying a powerset construction to the transition system $\mathscr{T} = \mathscr{T}_\chi$ over $R = R_\chi$. This construction is an adaption of the classical powerset construction for nondeterministic finite automata and can be seen as a one-agent variant of the powerset construction introduced by Reif (Reif 1984) for partial-information two-player games and its variant for CTL*K model checking (Bozzelli, Maubert, and Murano 2024). The acceptance condition in $\mathscr{D}_\chi$ is derived from the set $U = Sat_{\mathscr{T}}(\exists\varphi)$ in the basis of induction and $U = Sat_{\mathscr{T}}(\exists\varphi')$ in the step of induction.

Definition 4.3. *Let $\mathscr{T} = (S, \to, Init, R, L)$ with $R \subseteq P$, $\mathfrak{O} \subseteq P$ an observation set, and $U \subseteq S$. Then, $pow(\mathscr{T}, \mathfrak{O}, U)$ is the following DFA $\mathscr{D} = (\mathscr{X}_{\mathscr{D}}, \delta_{\mathscr{D}}, init_{\mathscr{D}}, F_{\mathscr{D}})$ over the alphabet 2^P. The state space is $\mathscr{X}_{\mathscr{D}} = 2^S \cup \{init_{\mathscr{D}}\}$. The transition function $\delta_{\mathscr{D}} : \mathscr{X}_{\mathscr{D}} \times 2^P \to \mathscr{X}_{\mathscr{D}}$ is given by (where $x \in 2^S$):*

$$\delta_{\mathscr{D}}(x, H) = \big\{ s' \in S : \text{ there exists } s \in x \text{ with } s \to s' \text{ and } L(s') \cap \mathfrak{O} = H \cap \mathfrak{O} \big\}$$

$$\delta_{\mathscr{D}}(init_{\mathscr{D}}, H) = \big\{ s \in Init : L(s) \cap \mathfrak{O} = H \cap \mathfrak{O} \big\}$$

The set of final states is $F_{\mathscr{D}} = \big\{ x \in \mathscr{X}_{\mathscr{D}} : x \cap U \neq \varnothing \big\}$.

We refer to $pow(\mathscr{T}, \mathfrak{O}, U)$ as the default history-DFA.

Example 4.4. Consider the transition system $\mathscr{T}_a$ depicted in Figure 3. To apply the construction from Definition 4.3, suppose $R = P = \{p\}$ and first that $\mathfrak{O} = \varnothing$ (as it is the case when constructing the default history-DFA for the steps semantics). Further, let $U = Sat_{\mathscr{T}_a}(\exists \bigcirc p) = \{s_0, s_2\}$. The reachable fragment of the default history-DFA $\mathscr{D}_1$ is depicted in the middle of Figure 3. If instead we assume $\mathfrak{O} = P$ (public-history semantics), the resulting default history-DFA is $\mathscr{D}_2$.

5 Complexity-Theoretic Results

At first glance the time complexity of our generic model checking algorithm is d-fold exponential for SLTL formulas φ where d is the maximal nesting depth of standpoint modalities in φ. This yields a nonelementary upper bound for the SLTL model checking problem.

For $* \in \{\text{step}, \text{pobs}, \text{public}, \text{decr}, \text{incr}\}$, let SLTL* denote SLTL under the semantics w.r.t. $\models_*$. We now study the complexity of the different SLTL* model checking problems.

Bounds on the Sizes of History-DFA and Time Bounds. Obviously, it suffices to construct the reachable fragment of the default history-DFA. Furthermore, we may apply a standard poly-time minimization algorithm to the default history-DFA. Thus, to establish better complexity bounds it suffices to provide elementary bounds on the sizes of minimal history-DFA for standpoint subformulas. Let us start with a simple observation on how to exploit the property that the history-DFA $\mathscr{D}_\chi$ are $\mathfrak{O}_\chi$-deterministic.

Remark 5.1. With the notations of the step of induction for $\chi = \langle\!\langle a \rangle\!\rangle \varphi$, the history-DFA $\mathscr{D}_\chi$ for χ is obtained by (the reachable fragment) of $pow(\mathscr{T}'_\chi, \mathfrak{O}_\chi, U_\chi)$ where $U_\chi = Sat_{\mathscr{T}_\chi}(\exists\varphi')$. Recall that $\mathscr{T}'_\chi$ is defined via a product construction $\mathscr{T}^R_a \bowtie \mathscr{D}_1 \bowtie \ldots \bowtie \mathscr{D}_k$ where the $\mathscr{D}_j$'s are history-DFAs for the maximal subformulas $\chi_j = \langle\!\langle b_j \rangle\!\rangle \psi_j$ of φ.

Suppose now that $i \in \{1, \ldots, k\}$ such that $\mathscr{D}_i$ is $\mathfrak{O}_\chi$-deterministic. Then, for all histories $h \in (2^P)^+$ and states $((s, O), x_1, \ldots, x_k)$ in $Reach(\mathscr{T}'_\chi, h)$ we have $x_i = \delta_i(init_i, h)$. Hence, if x is a reachable state in $\mathscr{D}$, say $x = \delta_{\mathscr{D}_\chi}(init_{\mathscr{D}}, h)$, then for all states $((s, O), x_1, \ldots, x_k) \in x$ we have $x_i = \delta_i(init_i, h)$. Thus, we may redefine the default history-DFA $\mathscr{D}_\chi$ as a product

$$pow(\mathscr{T}^R_a \bowtie \textstyle\prod_{j \neq i} \mathscr{D}_j, \mathfrak{O}_\chi) \bowtie \mathscr{D}_i$$

rather than a powerset construction of $\mathscr{T}^R_a \bowtie \mathscr{D}_1 \bowtie \ldots \bowtie \mathscr{D}_k$. From now on, we write $pow(\mathscr{T}, \mathfrak{O})$ for the structure $(\mathscr{X}_{\mathscr{D}}, \delta_{\mathscr{D}}, init_{\mathscr{D}})$ defined as in Definition 4.3. The acceptance condition $F_{\mathscr{D}_\chi}$ is (re)defined as the set of all states (x, x_i) in the above product (i.e., x is a subset of the state space of $\mathscr{T}^R_a \bowtie \prod_{j \neq i} \mathscr{D}_j$ and x_i a state in $\mathscr{D}_i$) such that $\{(\xi, x_i) : \xi \in x\} \cap U_\chi \neq \varnothing$. Let I denote the set of indices $i \in \{1, \ldots, k\}$ where the history-DFA for $\chi_i = \langle\!\langle b_i \rangle\!\rangle \psi_i$ is $\mathfrak{O}_\chi$-deterministic. W.l.o.g. $I = \{1, \ldots, \ell\}$. Then, we can think of $\mathscr{D}_\chi$ as (the reachable fragment of) a product

$$pow(\mathscr{T}^R_a \bowtie \mathscr{D}_{\ell+1} \bowtie \ldots \bowtie \mathscr{D}_k, \mathfrak{O}_\chi) \bowtie \mathscr{D}_1 \bowtie \ldots \bowtie \mathscr{D}_\ell.$$

If $\chi = \langle\!\langle a \rangle\!\rangle \varphi$ then $\mathfrak{O}_{\chi_i} = \mathfrak{O}_\chi$ for all maximal standpoint subformulas $\chi_i = \langle\!\langle b_i \rangle\!\rangle \psi_i$ of φ where $a = b_i$. Thus, by induction on $d = ad(\chi)$, Remark 5.1 yields that the size of the reachable fragment of the default history-DFA for a standpoint formula χ is d-fold exponentially bounded. Hence:

Lemma 5.2. *Under all five semantics, the time complexity of the algorithm of Section 4 is at most d-fold exponential when $d = ad(\phi)$.*

This d-fold exponential upper time bound will now be improved using further consequences of Remark 5.1.

We start with $\models_{\text{step}}$. Here, we have $\mathfrak{O}_\chi = \varnothing$ for all χ. With the notations used in Remark 5.1, the history-DFA $\mathscr{D}_1, \ldots, \mathscr{D}_k$ are $\mathfrak{O}_\chi$-deterministic. Thus, we can think of $\mathscr{D}_\chi$ as a product $pow(\mathscr{T}_a, \varnothing) \bowtie \mathscr{D}_1 \bowtie \ldots \bowtie \mathscr{D}_k$. But now, each of the $\mathscr{D}_i$'s also has this shape. In particular, if $\chi_i = \langle\!\langle a \rangle\!\rangle \psi_i$, then the projection of the first coordinate of the states and the transitions between them in the reachable fragment of $\mathscr{D}_i$ matches exactly those of $pow(\mathscr{T}_a, \varnothing)$. Thus, one can incorporate the information on final states and drop $pow(\mathscr{T}_a, \varnothing)$ from the product. As a consequence, the default history-DFA $\mathscr{D}_\chi$ can be redefined for $\models_{\text{step}}$ such that the state space of $\mathscr{D}_\chi$ is contained in $\prod_{b \in Ag(\chi)} 2^{S_b}$ where $Ag(\chi)$ denotes the set of agents $b \in Ag$ such that χ has a (possibly non-maximal) standpoint subformula of the form $\langle\!\langle b \rangle\!\rangle \psi$.

The situation is similar for $\models_{\text{public}}$ (where $\mathfrak{O}_\chi = P$ for all χ) and for $\models_{\text{decr}}$ (where $\mathfrak{O}_{\chi_i} \subseteq \mathfrak{O}_\chi$ for all standpoint subformulas χ_i of χ). In the case of $\models_{\text{public}}$ there are history-DFA where the state space is contained in $(\prod_{b\in Ag(\chi)} 2^{S_b}) \times 2^P$. For $\models_{\text{decr}}$, we assign history-DFA $\mathscr{D}_v$ to the nodes in the syntax tree of v that have the shape $\prod_w pow(\mathscr{T}_{a_w}, \mathfrak{O}_w^{\text{decr}})$ where w ranges over all nodes in the syntax subtree of v such that the formula given by w is a standpoint formula $\langle\!\langle a_w \rangle\!\rangle \varphi_w$.

Under the above mentioned simplification of the default-history DFAs, these observations yield:

Lemma 5.3. *For $* \in \{\text{step}, \text{public}, \text{decr}\}$, the algorithm of Section 4 for* SLTL* *model checking runs in (single) exponential time.*

Let us now look at $\models_{\text{incr}}$. Given a node v in the syntax tree of ϕ, let ϕ_v denote the subformula of ϕ that is represented by v and π_v the unique path $\pi_v = v_0 v_1 \ldots v_r$ in the syntax tree from the root v_0 to $v_r = v$, and let Ag_v denote the set of agents $b \in Ag$ such that ϕ_w has the form $\langle\!\langle b \rangle\!\rangle \psi_w$ for some $w \in \{v_1, \ldots, v_{r-1}\}$. By Remark 3.7, if $\phi_v = \langle\!\langle a \rangle\!\rangle \varphi$ then:

$$Q_v^{\text{incr}} = \bigcup_{b\in Ag_v} P_b \text{ and } \mathfrak{O}_v^{\text{incr}} = Q_v^{\text{incr}} \cup P_a$$

Let $\pi_v' = w_1 \ldots w_\ell$ be the sequence resulting from π_v when all nodes w where ϕ_w is not a standpoint subformula are removed. The observation sets along π_v' are increasing, i.e.,

$$\varnothing = \mathfrak{O}_{w_1}^{\text{incr}} \subseteq \mathfrak{O}_{w_2}^{\text{incr}} \subseteq \ldots \subseteq \mathfrak{O}_{w_\ell}^{\text{incr}} \subseteq P.$$

Thus, the number of indices $j \in \{2, \ldots, \ell\}$ where $\mathfrak{O}_{w_{j-1}}^{\text{incr}}$ and $\mathfrak{O}_{w_j}^{\text{incr}}$ are different is bounded by $N-1$ where $N = |Ag|$. If $\mathfrak{O}_{w_{j-1}}^{\text{incr}} = \mathfrak{O}_{w_j}^{\text{incr}}$, the history-DFA constructed for ϕ_{w_j} is $\mathfrak{O}_{w_{j-1}}^{\text{incr}}$-deterministic. We can now again apply Remark 5.1 to $\chi = \phi_{w_{j-1}}$ and $\chi_i = \phi_{w_j}$. This yields:

Lemma 5.4. *With $N = |Ag|$, our model checking algorithm for* SLTL$^{\text{incr}}$ *is N-fold exponentially time bounded.*

Space Bounds Lemma 3.9 shows that the SLTL$_d^{\text{pobs}}$ model checking problem is polynomially reducible to the LTLK$_d$ model checking problem under the perfect-recall semantics. The latter is known to be $(d-1)$-EXPSPACE-complete for $d \geqslant 2$ and PSPACE-complete for $d = 1$ (Bozzelli, Maubert, and Murano 2024). In combination with the known PSPACE lower bound for the LTL model checking problem (Sistla and Clarke 1985) and Lemmas 3.8 and 3.9 we obtain:

Corollary 5.5. *For $* \in \{\text{pobs}, \text{decr}, \text{incr}\}$, the* SLTL$_1^*$ *model checking problem is PSPACE-complete. For $d \geqslant 2$, the* SLTL$_d^{\text{pobs}}$ *model checking problem is in $(d-1)$-EXPSPACE.*

For the step and public-history semantics, the reduction provided in the proof of Lemma 3.9 can be adapted by dealing with an LTLK structure over a single agent. Thus, the generated LTLK formula has alternation depth at most 1.

Theorem 5.6. *For $* \in \{\text{step}, \text{public}\}$, the* SLTL* *model checking problem is polynomially reducible to the LTLK$_1$ model checking problem, and therefore PSPACE-complete.*

Lemma 3.9 and Theorem 5.6 show that the SLTL* model checking problem for $* \in \{\text{step}, \text{public}, \text{pobs}\}$ can be viewed as an instance of the LTLK model checking problem. An analogous statement holds for $\models_{\text{decr}}$ and $\models_{\text{incr}}$. The idea is to modify the reduction described in the proof of Lemma 3.9 by adding new agents for all standpoint subformulas $\chi = \langle\!\langle a \rangle\!\rangle \psi$ where $P_a \neq \mathfrak{O}_\chi$. This yields a polynomial reduction of the SLTL$_d^*$ model checking problem to the LTLK$_M$ model checking problem where $M = \min\{|Ag|, d\}$ and $* \in \{\text{incr}, \text{decr}\}$. From a complexity-theoretic view, this observation is irrelevant for the decremental semantics where we have PSPACE-completeness for alternation depth 1 (Corollary 5.5) and a single exponential upper time bound in the general case (Lemma 5.3). However, with the above mentioned results of (Bozzelli, Maubert, and Murano 2024), this observation improves the complexity-theoretic upper bound for the incremental semantics stated in Lemma 5.4:

Corollary 5.7. *The* SLTL$^{\text{incr}}$ *model checking problem is in $(N-1)$-EXPSPACE where $N = |Ag|$.*

6 Conclusion

We considered five different semantics for SLTL that differ in the amount of information the agents can extract from the history, and presented a generic model-checking algorithm applicable to all five semantics. The computational complexity of the algorithm, however, varies between the semantics due to the different numbers of necessary applications of the powerset construction. More precisely, the generic SLTL* model checking algorithm is m-fold exponentially time-bounded with $m = 1$ for $* \in \{\text{step}, \text{public}, \text{decr}\}$, $m = |Ag|$ for $* = \text{incr}$, and $m = ad(\phi)$ for $* = \text{pobs}$.

Algorithms with improved space complexity for $\models_{\text{step}}$, $\models_{\text{public}}$, $\models_{\text{pobs}}$, and $\models_{\text{incr}}$ are obtained via embeddings into LTLK. To match the space bounds, the model-checking algorithm of Section 5 can be adapted by combining classical on-the-fly automata-based LTL model checking techniques with an on-the-fly construction of history-DFA, similar to the techniques proposed in (Bozzelli, Maubert, and Murano 2024) for CTL*K. Analyzing if analogous techniques are applicable to obtain a polynomially space-bounded algorithm for $\models_{\text{decr}}$, providing lower bounds beyond PSPACE for $\models_{\text{incr}}$ and $\models_{\text{pobs}}$, as well as an experimental evaluation of the presented algorithm remain as future work. Further directions include extending the logic with common and distributed knowledge operators, as well as studying SLTL with past modalities such as *previous* and *since*.

Acknowledgments

The authors were supported by DFG grant 389792660 as part of TRR 248 (see https://perspicuous-computing.science), the Cluster of Excellence EXC 2050/1 (CeTI, project ID 390696704, as part of Germany's Excellence Strategy, and by BMBF (Federal Ministry of Education and Research) in DAAD project 57616814 (SECAI, School of Embedded and Composite AI) as part of the program Konrad Zuse Schools of Excellence in Artificial Intelligence. Joël Ouaknine is also affiliated with Keble College, Oxford as Emmy Network Fellow, and supported by ERC grant DynAMiCs (101167561).

References

Aghamov, R.; Baier, C.; Karimov, T.; Majumdar, R.; Ouaknine, J.; Piribauer, J.; and Spork, T. 2025. Model checking linear temporal logic with standpoint modalities. *arXiv preprint* arXiv:2502.20193.

Álvarez, L. G., and Rudolph, S. 2021. Standpoint Logic: Multi-Perspective Knowledge Representation. In F., N., and B., B., eds., *12th International Conference Frontiers in Artificial Intelligence and Applications (FOIS)*, volume 3344 of *Frontiers in Artificial Intelligence and Applications*, 3 – 17. IOS Press.

Alvarez, L. G.; Rudolph, S.; and Strass, H. 2022. How to agree to disagree. In Sattler, U.; Hogan, A.; Keet, M.; Presutti, V.; Almeida, J. P. A.; Takeda, H.; Monnin, P.; Pirrò, G.; and d'Amato, C., eds., *The Semantic Web – ISWC 2022*, 125–141. Cham: Springer International Publishing.

Alvarez, L. G. 2020. *Standpoint Logic: A Logic for Handling Semantic Variability, with Applications to Forestry Information*. Ph.D. Dissertation, University of Leeds.

Baier, C., and Katoen, J.-P. 2008. *Principles of Model Checking*. MIT Press.

Bozzelli, L.; Maubert, B.; and Murano, A. 2024. On the Complexity of Model Checking Knowledge and Time. *ACM Trans. Comput. Logic* 25(1).

Clarke, E. M.; Grumberg, O.; Kroening, D.; Peled, D.; and Veith, H. 2018. *Model Checking*. MIT Press. second edition.

Demri, S., and Walega, P. A. 2024. Computational complexity of standpoint LTL. In *27th European Conference on Artificial Intelligence (ECAI)*, volume 392 of *Frontiers in Artificial Intelligence and Applications*, 1206–1213. IOS Press.

Dima, C. 2009. Revisiting satisfiability and model-checking for CTLK with synchrony and perfect recall. In Fisher, M.; Sadri, F.; and Thielscher, M., eds., *Computational Logic in Multi-Agent Systems*, 117–131. Berlin, Heidelberg: Springer Berlin Heidelberg.

Emerson, E. A., and Lei, C. 1987. Modalities for model checking: Branching time logic strikes back. *Science of Computeer Programming* 8(3):275–306.

Emerson, E. A., and Sistla, A. P. 1984. Deciding full branching time logic. *Information and Control* 61(3):175–201.

Fagin, R.; Halpern, J. Y.; Moses, Y.; and Vardi, M. 2004. *Reasoning About Knowledge*. MIT Press.

Gigante, N.; Gómez Alvarez, L.; and Lyon, T. S. 2023. Standpoint Linear Temporal Logic. In *20th International Conference on Principles of Knowledge Representation and Reasoning*, 311–321.

Halpern, J. Y., and Vardi, M. Y. 1986. The complexity of reasoning about knowledge and time: Extended abstract. In Hartmanis, J., ed., *18th Annual ACM Symposium on Theory of Computing (STOC)*, 304–315. ACM.

Halpern, J. Y., and Vardi, M. Y. 1989. The complexity of reasoning about knowledge and time. i. lower bounds. *Journal of Computer and System Science* 38(1):195–237.

Halpern, J. Y. 1986. Reasoning about knowledge: An overview. In Halpern, J. Y., ed., *1st Conference on Theoretical Aspects of Reasoning about Knowledge*, 1–17. Morgan Kaufmann.

Kanellakis, P. C., and Smolka, S. A. 1990. CCS expressions, finite state processes, and three problems of equivalence. *Information and Computation* 86(1):43–68.

Reif, J. H. 1984. The complexity of two-player games of incomplete information. *Journal of Computer and System Science* 29(2):274–301.

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

van der Meyden, R., and Shilov, N. V. 1999. Model checking knowledge and time in systems with perfect recall. In Rangan, C. P.; Raman, V.; and Ramanujam, R., eds., *Foundations of Software Technology and Theoretical Computer Science*, 432–445. Berlin, Heidelberg: Springer Berlin Heidelberg.

Vardi, M. Y., and Wolper, P. 1986. An automata-theoretic approach to automatic program verification (preliminary report). In *Symposium on Logic in Computer Science (LICS)*, 332–344. IEEE Computer Society.

Vardi, M. Y., and Wolper, P. 1994. Reasoning about infinite computations. *Information and Computation* 115(1):1–37.