Set Semantics for Asynchronous TeamLTL: Expressivity and Complexity

Juha Kontinen
University of Helsinki, Finland

Max Sandström
University of Sheffield, UK
University of Helsinki, Finland

Jonni Virtema
University of Sheffield, UK
University of Helsinki, Finland

Abstract

We introduce and develop a set-based semantics for asynchronous TeamLTL. We consider two canonical logics in this setting: the extensions of TeamLTL by the Boolean disjunction and by the Boolean negation. We relate the new semantics with the original semantics based on multisets and establish one of the first positive complexity theoretic results in the temporal team semantics setting. In particular we show that both logics enjoy normal forms that can be utilised to obtain results related to expressivity and complexity (decidability) of the new logics.

2012 ACM Subject Classification Theory of computation → Logic and verification

Keywords and phrases Hyperproperties, Linear Temporal Logic, Team Semantics

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.60

Related Version *Full Version*: `https://arxiv.org/abs/2304.10915` [13]

Funding *Juha Kontinen*: Supported by the Academy of Finland grant 345634.
Max Sandström: Supported by the Academy of Finland grant 322795.
Jonni Virtema: Supported by the Academy of Finland grant 345634 and the DFG grant VI 1045/1-1.

1 Introduction

Linear temporal logic (LTL) is one of the most prominent logics for the specification and verification of reactive and concurrent systems. The core idea in model checking, as introduced in 1977 by Amir Pnueli [22], is to specify the correctness of a program as a set of infinite sequences, called traces, which define the acceptable executions of the system. In LTL-model checking one is concerned with trace sets that are definable by an LTL-formula. Ordinary LTL and its progeny are well suited for specification and verification of *trace properties*. These are properties of systems that can be checked by going through all executions of the system in isolation. A canonical example here is *termination*; a system terminates if each run of the system terminates. However not all properties of interest are trace properties. Many properties that are of prime interest, e.g., in information flow security, require a richer framework. The term *hyperproperty* was coined by Clarkson and Schneider [3] to refer to properties which relate multiple execution traces. A canonical example is *bounded termination*; one cannot check whether a system terminates in bounded time by only checking traces in isolation. Checking hyperproperties is vital in information flow security where dependencies between secret inputs and publicly observable outputs of a system are considered potential security violations. Commonly known properties of that type are noninterference [24, 20] and observational determinism [30]. Hyperproperties are not limited to the area of information flow control; e.g., distributivity and other system properties like fault tolerance can be expressed as hyperproperties [5].

During the past decade, the need for being able to formally specify hyperproperties has led to the creation of families of new logics for this purpose, since LTL and other established temporal logics can only specify trace properties. The two main families of the new logics are the so-called *hyperlogics* and logics that adopt *team semantics*. In the former approach temporal logics such as LTL, computation tree logic (CTL), and quantified propositional temporal logic (QPTL) are extended with explicit trace and path quantification, resulting in logics like HyperLTL [2], HyperCTL* [2], and HyperQPTL [23, 4]. The latter approach (which we adopt here) is to lift the semantics of temporal logics to sets of traces directly by adopting team semantics yielding logics such as TeamLTL [15, 7] and TeamCTL [14, 7].

Krebs et al. [15] introduced two versions of LTL with team semantics: a synchronous semantics and an asynchronous variant that differ on how the evolution of time is linked between computation traces when temporal operators are evaluated. In the synchronous semantics time proceeds in lock-step, while in the asynchronous variant time proceeds independently on each trace. For example the formula "F terminate" (here F denotes the future-operator and "terminate" is a proposition depicting that a trace has terminated) defines the hyperproperty "bounded termination" under synchronous semantics, while it expresses the trace property "termination" under asynchronous semantics. The elegant definition of bounded termination exemplifies one of the main distinguishing factors of team logics from hyperlogics; namely the ability to refer directly to unbounded number of traces. Each hyperlogic-formula has a fixed number of trace quantifiers that delineate the traces involved in the evaluation of the formula. Another distinguishing feature of team logics lies in their ability to enrich the logical language with novel atomic formulae for stating properties of teams. The most prominent of these are the *dependence atom* $\mathrm{dep}(\bar{x}, \bar{y})$ (stating that the values of the variables $\bar{x}$ functionally determine the values of $\bar{y}$) and *inclusion atom* $\bar{x} \subseteq \bar{y}$ (expressing the inclusion dependency that all the values occurring for $\bar{x}$ must also occur as a value for $\bar{y}$).

As an example, let $o_1, \ldots, o_n$ be public observables and assume that c reveals confidential information. The atom $(o_1, \ldots o_n, c) \subseteq (o_1, \ldots o_n, \neg c)$ expresses a form of non-inference by stating that an observer cannot infer the value of the confidential bit from the outputs.

While HyperLTL and other hyperlogics have been studied extensively, many of the basic properties of TeamLTL are still not well understood. Krebs et al.[15] showed that synchronous TeamLTL and HyperLTL are incomparable in expressivity and that the asynchronous variant collapses to LTL. Not much was known about the complexity aspects of TeamLTL until Lück [18] showed that the complexity of satisfiability and model checking of synchronous TeamLTL with Boolean negation $\sim$ is equivalent to the decision problem of third-order arithmetic. Subsequently, Virtema et al. [29] embarked for a more fine-grained analysis of the complexity of synchronous TeamLTL and discovered a decidable syntactic fragment (the so-called *left-flat fragment*) and established that already a very weak access to the Boolean negation suffices for undecidability. They also showed that synchronous TeamLTL and its extensions can be translated to HyperQPTL$^+$, which is an extension of HyperLTL by (non-uniform) quantification of propositions. Kontinen and Sandström [12] defined translations between extensions of TeamLTL and the three-variable fragment of first-order team logic to utilize the better understanding of first-order team semantics. They also showed that any logic effectively residing between synchronous TeamLTL extended with the Boolean negation and second-order logic inherits the complexity properties of the extension of TeamLTL with the Boolean negation. Finally, Gutsfeld et al. [7] reimagined the setting of temporal team semantics to be able to model richer forms of (a)synchronicity by developing the notion of time-evaluation functions. In addition to reimagining the framework, they discovered decidable logics which however relied on restraining time-evaluation functions to be either *k-context-bounded* or *k-synchronous*. It is worth noting that recently asynchronous hyperlogics have been considered also in several other articles (see, e.g., [8, 1]).

Almost all complexity theoretic results previously obtained for TeamLTL have been negative, and the few positive results have required drastic restrictions in syntax or semantics. In this article we take a take a fresh look at expressive extensions of asynchronous TeamLTL. Recent works on synchronous TeamLTL have revealed that quite modest extensions of synchronous TeamLTL are undecidable. Thus, our study of asynchronous TeamLTL partly stems from our desire to discover decidable but expressive logics for hyperproperties. Until now, all the papers on temporal team semantics have explicitly or implicitly adopted a semantics based on multisets of traces. In the team semantics literature, this often carries the name *strict semantics*, in contrast to *lax semantics* which is de-facto set-based semantics. In database theory, it is ubiquitous that tasks that are computationally easy under set based semantics become untractable in the multiset case. In the team semantics setting this can be already seen in the model checking problem of propositional inclusion logic PL($\subseteq$) which is P-complete under lax semantics, but NP-complete under strict semantics [10]. Our new set-based framework offers a setting that drops the accuracy that accompanies adoption of multiset semantics in favour of better computational properties. Consider the following formula expressing a form of strong non-inference in parallel computation: $\mathsf{G}((o_1, ..., o_n, c) \subseteq (o_1, ..., o_n, \neg c))$, where $o_1, ..., o_n$ are observable outputs and c is confidential. In the synchronous setting, the formula expresses that during a synchronous computation, at any given time, an observer cannot infer the value of the secret c from the outputs. In the asynchronous setting, the formula states a stronger property that the above property holds for all computations (not only synchronous). In the multiset setting the number of parallel computation nodes is fixed, while in the new lax semantics, we drop that restriction, and consider an undefined number of computation nodes. The condition is stronger in lax semantics; and intuitively easier to falsify, which makes model checking in practice easier.

Our contribution. We introduce and develop a set-based semantics for asynchronous TeamLTL, which we name *lax semantics* and write TeamLTL^l. We consider two canonical logics in this setting: the extensions of TeamLTL^l by the Boolean disjunction TeamLTL$^l(\otimes)$ and by the Boolean negation TeamLTL$^l(\sim)$. By developing the basic theory of lax asynchronous TeamLTL, we discover some fascinating connections between the strict and lax semantics. We discover that both of the logics enjoy normal forms that can be utilised to obtain expressivity and complexity results. Tables 1 and 2 summarise our results. For comparison, Table 3 summarises the known results on complexity of synchronous TeamLTL.

2 Preliminaries

Fix a set AP of *atomic propositions*. The set of formulae of LTL (over AP) is generated by the grammar: $\varphi ::= p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bigcirc \varphi \mid \mathsf{G}\,\varphi \mid \varphi\, \mathsf{U}\, \varphi$, where $p \in$ AP. We adopt the convention that formulae are given in negation normal form, i.e., $\neg$ is allowed only in front of atomic propositions. Note that this is an expressively complete set of LTL-formulae. The logical constants $\top, \bot$ and the operators F and W can be defined in the usual way: $\bot := p \wedge \neg p$, $\top := p \vee \neg p$, $\mathsf{F}\,\varphi := \top\,\mathsf{U}\,\varphi$, and $\varphi\,\mathsf{W}\,\psi := (\varphi\,\mathsf{U}\,\psi) \vee \mathsf{G}\,\varphi$. Note also the equivalences $\neg \bigcirc \varphi \equiv \bigcirc \neg\varphi$, $\neg\,\mathsf{F}\,\varphi \equiv \mathsf{G}\,\neg\varphi$, and $\neg(\varphi\,\mathsf{U}\,\psi) \equiv (\neg\varphi\,\mathsf{W}(\neg\psi \wedge \neg\varphi))$.

A *trace* t over AP is an infinite sequence from $(2^{\mathrm{AP}})^\omega$. For a natural number $i \in \mathbb{N}$, we denote by $t[i]$ the $(i+1)$th letter of t and by $t[i, \infty]$ the postfix $(t[j])_{j \geq i}$ of t. Semantics of LTL is defined in the usual manner (see e.g., [21]). For example, $t \models p$ iff $p \in t[0]$ and $t \models \bigcirc \varphi$ iff $t[1, \infty] \models \varphi$. The *truth value* of a formula φ on a trace t is denoted by $[\![\varphi]\!]_t \in \{0, 1\}$.

Table 1 Expressivity hierarchy of the asynchronous logics considered in the paper. Logics with lax or strict semantics are here referred with the superscripts l and s, respectively. For the definitions of left flatness, quasi flatness, and left downward closure, we refer to Definitions 7 and 13. †: This follows since only TeamLTL$^l(\ovee)$ is downward closed (cf. Theorem 8 and Definition 13). Theorem 8 implies that for TeamLTL($\sim$)-formulae in quasi-flat form the strict and lax semantics coincide.

TeamLTL$^{s/l}$		left-flat–TeamLTL$^s(\ovee)$	$\overset{\text{Cor.12}}{<}$	TeamLTL$^s(\ovee)$
$\wedge$ Ex. 6		$\vert\vert\vert$ Thm. 8		
TeamLTL$^l(\ovee)$	$\overset{\text{Thm. 10}}{\equiv}$	left-flat–TeamLTL$^l(\ovee)$	$\overset{\dagger}{<}$	quasi-flat–TeamLTL$^{s/l}(\sim)$
				$\vert\vert\vert$ Thm. 14
				left-dc–TeamLTL$^l(\sim)$

Table 2 Complexity results of this paper. All results are completeness results if not otherwise specified. PL($\sim$) refers to the propositional fragment of TeamLTL($\sim$) which embeds also to left-dc-TeamLTL$^l(\sim)$. †: All PSPACE-completeness results for satisfiability in strict semantics and TeamLTL^l follow directly from classical LTL by downward closure and singleton equivalence similar to [15, Proposition 5.4]. ATIME-ALT(exp, poly) refers to alternating exponential time with polynomially many alternations while TOWER(poly) refers to problems that can be decided by a deterministic TM in time bounded by an exponential tower of 2's of polynomial height.

Logic (asynchronous semantics)	Complexity of: model checking	Complexity of: satisfiability	References
LTL	PSPACE	PSPACE	[25]
PL($\sim$)	ATIME-ALT(exp, poly)	ATIME-ALT(exp, poly)	[9]
TeamLTL$^{l/s}$	PSPACE	PSPACE	[15], Theorem 5
left-flat-TeamLTL$^{s/l}(\ovee)$	PSPACE	PSPACE	Theorem 17
TeamLTL$^l(\ovee)$	PSPACE	PSPACE	Theorem 17
TeamLTL$^s(\ovee)$	???	PSPACE	†
TeamLTL^s(dep)	NEXPTIME-hard	PSPACE	[15]
left-dc-TeamLTL$^l(\sim)$	in TOWER(poly)	in TOWER(poly)	Theorem 17

Table 3 Complexity results for synchronous strict semantics. All results are completeness results if not otherwise specified. †: All PSPACE-completeness results for satisfiability follow directly from classical LTL by downward closure and singleton equivalence similar to [15, Proposition 5.4].

Logic (sync. strict semantics)	Complexity of: model checking	Complexity of: satisfiability	References
TeamLTL	PSPACE	PSPACE	[15]
left-flat-TeamLTL($\ovee$)	in EXPSPACE	PSPACE	[29]
TeamLTL(dep)	NEXPTIME-hard	PSPACE	[15]
TeamLTL($\ovee$)	???	PSPACE	†
TeamLTL($\ovee, \subseteq$)	Σ_1^0-hard	Σ_1^0-hard	[29]
TeamLTL($\sim$)	third-order arithmetic	third-order arithmetic	[18]

Next we present the so-called asynchronous team semantics for LTL introduced in [15]. In [15], the release operator was defined slightly erroneously; we fix the issue here by taking G as primitive and defining R using G and U. Informally, a multiset of traces T is a collection of traces with possible repetitions. Formally, we represent T as a set of pairs (i, t), where i is an *index* (from some suitable large set) and t is a trace. We stipulate that the elements of a multiset have distinct indices. From now on, we will always omit the index and write t instead of (i, t). For multisets T and S, $T \uplus S$ denotes the disjoint union of T and S (obtained by stipulating that traces in S and T have disjoint sets of indices). Note that all the functions f with domain T are actually of the form $f((i, t))$ and may map different copies of the trace t differently. A *team* (*multiteam*, resp.) is a set (multiset, resp.) of traces. If $f : T \to \mathbb{N}$ is a function, we define the updated team $T[f, \infty] := \{t[f(t), \infty] \mid t \in T\}$, where f determines for each trace a point in time it updates to. For functions f and f' as above, we write $f' < f$, if $f'(t) < f(t)$ for all $t \in T$. The underlying team $\mathrm{support}(T) := \{t \mid (i, t) \in T\}$ of a multiteam T is called the *support* of T.

▶ **Definition 1** (Team Semantics for LTL). *Let T be a multiteam, and φ and ψ LTL-formulae. The asynchronous team semantics of TeamLTL is defined as follows.*

$$
\begin{array}{lcl}
T \models l & \Leftrightarrow & t \models l \text{ for all } t \in T, \text{ where } l \in \{p, \neg p \mid p \in \mathrm{AP}\} \text{ is a literal} \\
T \models \varphi \wedge \psi & \Leftrightarrow & T \models \varphi \text{ and } T \models \psi \\
T \models \varphi \vee \psi & \Leftrightarrow & \exists T_1, T_2 \text{ s.t. } T_1 \uplus T_2 = T \text{ and } T_1 \models \varphi \text{ and } T_2 \models \psi \\
T \models \bigcirc \varphi & \Leftrightarrow & T[1, \infty] \models \varphi, \text{ where } 1 \text{ is the constant function } t \mapsto 1 \\
T \models \mathsf{G}\varphi & \Leftrightarrow & \forall f : T \to \mathbb{N} \quad T[f, \infty] \models \varphi \\
T \models \varphi \mathsf{U} \psi & \Leftrightarrow & \exists f : T \to \mathbb{N} \quad T[f, \infty] \models \psi \text{ and } \forall f' < f : T'[f', \infty] \models \varphi, \\
& & \text{where } T' := \{t \in T \mid f(t) \neq 0\}
\end{array}
$$

The synchronous variant of the semantics is obtained by allowing f to range only over constant functions. We take the asynchronous semantics as the standard semantics and write TeamLTL *for asynchronous* TeamLTL.

We also consider the *Boolean disjunction* $\varovee$ and *Boolean negation* $\sim$ interpreted as usual: $T \models \varphi \varovee \psi$ iff ($T \models \varphi$ or $T \models \psi$), and $T \models {\sim}\varphi$ iff $T \not\models \varphi$.

Next we define some important semantic properties of formulae studied in the literature. A logic has one of the properties if every formula of the logic has the property. It is easy to check that TeamLTL has all the properties listed [15] whereas its extension with the Boolean disjunction has all but flatness and the extension with Boolean negation has none.

(Downward closure) If $T \models \varphi$ and $S \subseteq T$, then $S \models \varphi$.
(Empty team property) $\emptyset \models \varphi$.
(Flatness) $T \models \varphi$ iff $\{t\} \models \varphi$ for all $t \in T$.
(Singleton equivalence) $\{t\} \models \varphi$ iff $t \models \varphi$.

We will now justify our choice of semantics. The semantic rules for literals, conjunction, and disjunction are the standard ones in team semantics, and which have been motivated numerous times in the literature [26]. Two important properties for the logic to have, for it to be a conservative extension of LTL, are flatness and singleton equivalence. These properties also motivated the original definition of asynchronous TeamLTL [15]. The given semantics for $\bigcirc$ is the only possible one that satisfies flatness. The same is true for F (i.e., $\top \mathsf{U} \varphi$) and G; moreover the semantics clearly capture the intuitive meanings of asynchronously in the future and asynchronously globally, respectively. The given semantics for U preserves flatness and singleton equivalence, and adequately captures the intuitive meaning of asynchronous until. The framework of asynchronous TeamLTL then allows us to define different variants of the familiar temporal operators. E.g., $\varphi \,\mathsf{W}_1\, \psi := \mathsf{G}\,\varphi \vee \varphi \,\mathsf{U}\, \psi$ and $\varphi \,\mathsf{W}_2\, \psi := \mathsf{G}\,\varphi \otimes \varphi \,\mathsf{U}\, \psi$ define different variants of *weak until*; the first of which is flat, while the second is not.

$$T \models \varphi \,\mathsf{W}_1\, \psi \quad \Leftrightarrow \quad \exists T_1, T_2 \text{ s.t. } T_1 \uplus T_2 = T,\ T_1 \models \mathsf{G}\,\varphi \text{ and } T_2 \models \varphi \,\mathsf{U}\, \psi$$

$$T \models \varphi \,\mathsf{W}_2\, \psi \quad \Leftrightarrow \quad T \models \mathsf{G}\,\varphi \text{ or } T \models \varphi \,\mathsf{U}\, \psi$$

Similarly $\varphi \,\mathsf{R}_1\, \psi := \psi \,\mathsf{U}((\psi \wedge \varphi) \vee \mathsf{G}\,\psi)$ and $\varphi \,\mathsf{R}_2\, \psi := \psi \,\mathsf{U}((\psi \wedge \varphi) \otimes \mathsf{G}\,\psi)$ give rise to different variants of *release*. Moreover, with $\sim$ one can define additional dual operators.

A defining feature of team semantics is the ability to enrich logics with novel atomic statements describing properties of teams in a modular fashion. For example, *dependence atoms* $\mathrm{dep}(\varphi_1, \dots, \varphi_n, \psi)$ and *inclusion atoms* $\varphi_1, \dots, \varphi_n \subseteq \psi_1, \dots, \psi_n$, with $\varphi_1, \dots, \varphi_n, \psi, \psi_1, \dots, \psi_n$ being LTL-formulae, have been studied extensively in first-order and modal team semantics. The dependence atom states that the truth value of ψ is functionally determined by that of $\varphi_1, \dots, \varphi_n$ whereas the inclusion atom states that each value combination of $\varphi_1, \dots, \varphi_n$ must also occur as a value combination for $\psi_1, \dots, \psi_n$. Formally:

$$T \models \mathrm{dep}(\varphi_1, \dots, \varphi_n, \psi) \;\text{ iff }\; \forall t, t' \in T : \Big(\bigwedge_{1 \le j \le n} [\![\varphi_j]\!]_t = [\![\varphi_j]\!]_{t'} \Big) \Rightarrow [\![\psi]\!]_t = [\![\psi]\!]_{t'}$$

$$T \models \varphi_1, \dots, \varphi_n \subseteq \psi_1, \dots, \psi_n \;\text{ iff }\; \forall t \in T\, \exists t' \in T : \bigwedge_{1 \le j \le n} [\![\varphi_j]\!]_t = [\![\psi_j]\!]_{t'}$$

Consider the following exemplary formula: $\mathsf{G}\,\mathrm{dep}(i_1, i_2, o) \vee \mathsf{G}\,\mathrm{dep}(i_2, i_3, o)$. The formula states that the executions of the system can be decomposed into two parts; in the first part, the output o is determined by the inputs i_1 and i_2, and in the second part, o is determined by the inputs i_2 and i_3.

If $\mathcal{A}$ is a collection of atoms and connectives, TeamLTL($\mathcal{A}$) denotes the extension of TeamLTL with the atoms and connectives in $\mathcal{A}$. It is straightforward to see (in analogy to the modal team semantics setting [11]) that any dependency such as the ones above is determined by a finite set of n-ary Boolean relations. Let B be a set of n-ary Boolean relations. We define the property $[\varphi_1, \dots, \varphi_n]_B$ for an n-tuple $(\varphi_1, \dots, \varphi_n)$ of LTL-formulae:

$$T \models [\varphi_1, \dots, \varphi_n]_B \quad \text{iff} \quad \{([\![\varphi_1]\!]_t, \dots, [\![\varphi_n]\!]_t) \mid t \in T\} \in B.$$

Expressions of the form $[\varphi_1, \dots, \varphi_n]_B$ are *generalised atoms*. It was shown in [29] that, in the synchronous setting, TeamLTL($\sim$) is expressively complete with respect to all generalised atoms, whereas the extension of TeamLTL($\otimes$) with the so-called *flattening operator* can express any downwards closed generalised atoms. These results readily extend to the asynchronous setting. Moreover the flattening operator renders itself unnecessary due to flatness of asynchronous TeamLTL. The results imply, e.g, that the (downwards closed) dependence atoms can be expressed in both of the logics TeamLTL($\sim$) and TeamLTL($\otimes$), and inclusion atoms in turn are expressible in TeamLTL($\sim$). The proof of the following theorem is essentially the same as the proof of [28, Proposition 17]. Below $L \equiv L'$ denotes the equiexpressivity of the logics L and L'.

▶ **Theorem 2.** *Let $\mathcal{A}$, $\mathcal{D}$ be the sets of all generalised atoms, and all downward closed generalised atoms. Then* TeamLTL($\mathcal{D}, \otimes$) $\equiv$ TeamLTL($\otimes$) *and* TeamLTL($\mathcal{A}, \sim$) $\equiv$ TeamLTL($\sim$).

3 Set-based semantics for TeamLTL

Next we define a relaxed version of the asynchronous semantics. We call it *lax* semantics as it corresponds to the so-called lax semantics of first-order team semantics (see e.g., [6]). From now on we refer to the semantics of Definition 1 as strict semantics. The possibility of considering lax semantics for TeamLTL was suggested by Lück already in [19] but the full definition was not given. Intuitively, lax semantics can always be obtained from a strict one by checking what strict semantics would yield if multiteams were enriched with unbounded many copies of each of its traces. One of the defining features of lax semantics is that it is unable to distinguish multiplicities, which is formalised by Proposition 4 below.

We need some notation for the new definition. We write $\mathcal{P}(\mathbb{N})^+$ to denote $\mathcal{P}(\mathbb{N}) \setminus \{\emptyset\}$. For a team T and function $f\colon T \to \mathcal{P}(\mathbb{N})^+$, we set $T[f,\infty] := \{t[s,\infty] \mid t \in T, s \in f(t)\}$. For $T' \subseteq T$, $f\colon T \to \mathcal{P}(\mathbb{N})^+$, and $f'\colon T' \to \mathcal{P}(\mathbb{N})^+$, we define that $f' < f$ if and only if

$$\forall t \in T'\colon \min(f'(t)) \leq \min(f(t)) \text{ and, if } \max(f(t)) \text{ exists, } \max(f'(t)) < \max(f(t)).$$

▶ **Definition 3** ($\mathrm{TeamLTL}^l$). *Let T be a team, and φ and ψ TeamLTL-formulae. The lax semantics is defined as follows. We only list the cases that differ from the strict semantics.*

$$\begin{aligned}
T \models^l \varphi \vee \psi \quad &\Leftrightarrow \quad \exists T_1, T_2 \text{ s.t. } T_1 \cup T_2 = T \text{ and } T_1 \models \varphi \text{ and } T_2 \models \psi\\
T \models^l \mathsf{G}\,\varphi \quad &\Leftrightarrow \quad \forall f\colon T \to \mathcal{P}(\mathbb{N})^+ \text{ it holds that } T[f,\infty] \models^l \varphi\\
T \models^l \varphi\,\mathsf{U}\,\psi \quad &\Leftrightarrow \quad \exists f\colon T \to \mathcal{P}(\mathbb{N})^+ \text{ such that } T[f,\infty] \models^l \psi \text{ and}\\
&\qquad \forall f'\colon T' \to \mathcal{P}(\mathbb{N})^+ \text{s.t. } f' < f, \text{ it holds that } T'[f',\infty] \models^l \varphi \text{ or } T' = \emptyset,\\
&\qquad \text{where } T' := \{t \in T \mid \max(f(t)) \neq 0\}
\end{aligned}$$

In the context we will be considering in this article, the subformulae φ in the definition of the until operator U always have the empty team property and thus we disregard the possibility that the team T' is empty in our proofs, as that case follows from the empty team property.

The above set-based semantics can also be viewed in terms of multisets. In that case functions f are quantified uniformly, i.e. we restrict our consideration to functions where $f(i,t) = f(j,t)$. Furthermore, the semantics for disjunction is defined in a way that omits references to multiplicities. In order to relate our new logics to the old multiteam based ones, we extend the lax semantics to multiteams T by stipulating that $T \models^l \varphi$ iff $\mathrm{support}(T) \models^l \varphi$.

The following proposition shows that $\mathrm{TeamLTL}^l(\sim)$ satisfies the so-called *locality* property, see full version of this article [13] for the proof. For a trace t over AP' and $\mathrm{AP} \subseteq \mathrm{AP}'$, the *reduction of* t to AP, $t_{\restriction \mathrm{AP}}$, is a sequence from $(2^{\mathrm{AP}})^\omega$ such that $p \in t[i]$ if and only if $p \in t_{\restriction \mathrm{AP}}[i]$, for all $p \in \mathrm{AP}$ and $i \in \mathbb{N}$. For a team T over AP' we define the *reduction of* T to AP by $T_{\restriction \mathrm{AP}} = \{t_{\restriction \mathrm{AP}} \mid t \in T\}$.

▶ **Proposition 4.** *Let T be a team and φ a $\mathrm{TeamLTL}^l(\sim)$-formula with variables in* AP. *Now $T \models^l \varphi$ iff $T_{\restriction \mathrm{AP}} \models^l \varphi$.*

The next theorem displays that lax semantics enjoys the same fundamental properties as its strict counterpart. The proof via a straightforward induction, see full version of this article [13] for details.

▶ **Theorem 5.** $\mathrm{TeamLTL}^l$ *satisfies downward closure, empty team property, singleton equivalence, and flatness.*

The following example establishes that the new lax semantics differs from the strict semantics, and that in the old semantics multiplicities matter. Moreover, we obtain $\mathrm{TeamLTL}^l < \mathrm{TeamLTL}^l(\circledvee)$ by showing that the latter is not flat.

▶ **Example 6.** Let φ be the formula $\mathsf{G}(p \ovee q)$, $T_1 := \{t\}$ and $T_2 := \{(1,t),(2,t)\}$, where $t := \{p\}\{q\}^\omega$. It is easy to check that $T_1 \models \varphi$ but $T_1 \not\models^l \varphi$ (which is witnessed by $T[f,\infty] \not\models^l p \ovee q$ for $f(t) := \{0,1\}$). Likewise $T_2 \not\models \varphi$. Moreover $\{s_i\} \models^l \varphi$, for $i \in \{1,2\}$, but $\{s_1,s_2\} \not\models^l \varphi$, where $s_1 := \{p\}^\omega$ and $s_2 := \{q\}^\omega$.

We will also consider the following fragments of TeamLTL($\ovee$) and TeamLTL($\sim$).

▶ **Definition 7.** *A formula φ of* TeamLTL($\ovee$) *is called* left-flat, *if in all of its subformulae of the form $\mathsf{G}\,\psi$ and $\psi \,\mathsf{U}\, \theta$, the subformula ψ is an* LTL*-formula. A formula φ of* TeamLTL($\sim,\ovee$) *is called* left-downward closed, *if in all of its subformulae of the form $\mathsf{G}\,\psi$ and $\psi \,\mathsf{U}\, \theta$, the subformula ψ is an* TeamLTL($\ovee$)*-formula.*

We will later show that the above syntactic restriction for flatness could be replaced by a semantic restriction (see Corollary 11). The proof of the following theorem is in the full version of this article [13].

▶ **Theorem 8.** *For all $\varphi \in$* TeamLTL^l($\ovee$) *the following two claims hold:*
1. *φ is downward closed and has the empty team property, and*
2. *if φ is left-flat, then $T \models \varphi$ iff* support$(T) \models^l \varphi$ *for all multiteams T.*

The restriction to left-flat formulae in case (2) above is necessary by Example 6.

4 Normal Forms for TeamLTL with Boolean Disjunction and Negation

In this section we develop normal forms for our logics, which we then utilise to obtain strong expressivity and complexity results.

▶ **Definition 9.** *A formula φ is in $\ovee$-disjunctive normal form if it is of the form $\bigovee_{i\in I} \alpha_i$, where α_i are* LTL*-formulae.*

Every formula of TeamLTL^l($\ovee$) can be transformed into an equivalent $\ovee$-disjunctive normal form. This result is similar to the one proved in [27] for team-based modal logic ML($\ovee$). In the following $|\varphi|$ denotes the the length of the formula φ.

▶ **Theorem 10.** *Every $\varphi \in$* TeamLTL^l($\ovee$) *is logically equivalent to a formula $\varphi^* = \bigovee_{i\in I} \alpha_i$ in $\ovee$-disjunctive normal form, where $|\alpha_i| \leq |\varphi|$ and $|I| = 2^k$, where k is the number of $\ovee$ in φ.*

Proof. The proof proceeds by induction on the structure of formulae. Note that atomic formulae are already in the normal form and that the case for $\ovee$ is trivial. The remaining cases are defined as follows:

$$(\psi \wedge \theta)^* := \bigovee_{i\in I, j\in J} (\alpha_i^\psi \wedge \alpha_j^\theta) \qquad (\psi \vee \theta)^* := \bigovee_{i\in I, j\in J} (\alpha_i^\psi \vee \alpha_j^\theta)$$

$$(\bigcirc\,\psi)^* := \bigovee_{i\in I} \bigcirc\,\alpha_i^\psi \qquad (\mathsf{G}\,\psi)^* := \bigovee_{i\in I} \mathsf{G}\,\alpha_i^\psi$$

$$(\psi \,\mathsf{U}\, \theta)^* := \bigovee_{i\in I, j\in J} (\alpha_i^\psi \,\mathsf{U}\, \alpha_j^\theta).$$

where α_i^ψ and α_j^θ are the flat formulae in the disjunctive normal forms of ψ and θ respectively, and I and J are the respective index sets.

Suppose $\varphi = \psi \wedge \theta$ and that $\psi \equiv \bigovee_{i\in I} \alpha_i^\psi$ and $\theta \equiv \bigovee_{i\in J} \alpha_j^\theta$ (induction hypothesis). Now $T \models^l \varphi$ if and only if $T \models^l \psi$ and $T \models^l \theta$. The latter holds, if and only if $T \models^l \alpha_k^\psi$ and $T \models^l \alpha_{k'}^\theta$, for some k and k'. This can be equivalently expressed as $T \models^l \bigovee_{i,j} (\alpha_i^\psi \wedge \alpha_j^\theta)$, i.e. $T \models^l \varphi^*$.

Suppose $\varphi = \psi \vee \theta$ and that $\psi \equiv \bigovee_{i\in I} \alpha_i^\psi$ and $\theta \equiv \bigovee_{i\in J} \alpha_j^\theta$. By definition $T \models^l \varphi$ if and only if there exists $T' \cup T'' = T$ such that $T' \models^l \psi$ and $T'' \models^l \theta$. By the induction hypothesis the latter is equivalent with $T' \models^l \bigovee_{i\in I} \alpha_i^\psi$ and $T'' \models^l \bigovee_{j\in J} \alpha_j^\theta$. By definition this holds if and only if there are k' and k'' such that $T' \models^l \alpha_{k'}^\psi$ and $T'' \models^l \alpha_{k''}^\theta$, which is equivalent with $T \models^l \alpha_{k'}^\psi \vee \alpha_{k''}^\theta$ for some k' and k'', by definition. Equivalently then $T \models^l \bigovee_{i\in I, j\in J}(\alpha_i^\psi \vee \alpha_j^\theta)$.

Suppose $\varphi = \bigcirc\psi$ and that $\psi \equiv \bigovee_{i\in I} \alpha_i^\psi$. By definition $T \models^l \varphi$ is equivalent with $T[1,\infty] \models^l \psi$. By the induction hypothesis the latter holds if and only if $T[1,\infty] \models^l \bigovee_{i\in I} \alpha_i^\psi$, which by definition is equivalent with $T[1,\infty] \models^l \alpha_k^\psi$ for some $k \in I$. The latter holds if and only if $T \models^l \bigcirc\alpha_k^\psi$ for some $k \in I$, which is equivalent with $T \models^l \bigovee_{i\in I} \bigcirc\alpha_i^\psi$.

Suppose $\varphi = \mathsf{G}\,\psi$ and that $\psi \equiv \bigovee_{i\in I} \alpha_i^\psi$. Suppose that $T \models^l \varphi$. By definition for all functions $f\colon T \to \mathcal{P}(\mathbb{N})^+$ it holds that $T[f,\infty] \models^l \psi$. By the induction hypothesis $T[f,\infty] \models^l \bigovee_{i\in I} \alpha_i^\psi$ for all f. Especially this holds for the total function defined for every $t \in T$ by $f_{max}(t) := \mathbb{N}$. Thus $T[f_{max},\infty] \models^l \alpha_k^\psi$ for some k. By downward closure it holds that $T[f',\infty] \models^l \alpha_k^\psi$ for all $f'\colon T \to \mathcal{P}(\mathbb{N})^+$. Hence $T \models^l \mathsf{G}\,\alpha_k^\psi$, and thus $T \models^l \bigovee_{i\in I} \mathsf{G}\,\alpha_i^\psi$. The other direction is analogous.

Suppose $\varphi = \psi \,\mathsf{U}\, \theta$ and that $\psi \equiv \bigovee_{i\in I} \alpha_i^\psi$ and $\theta \equiv \bigovee_{j\in J} \alpha_j^\theta$. Suppose $T \models^l \varphi$. By definition there exists a function $f\colon T \to \mathcal{P}(\mathbb{N})^+$ such that $T[f,\infty] \models^l \theta$ and for all functions $f'\colon T' \to \mathcal{P}(\mathbb{N})^+$ such that $f' < f$, $T'[f',\infty] \models^l \psi$, where $T' := \{t \in T \mid f(t) \neq 0\}$. Hence by the induction hypothesis $T[f,\infty] \models^l \bigovee_{j\in J} \alpha_j^\theta$, which is equivalent with $T[f,\infty] \models^l \alpha_k^\theta$ for some $k \in J$, and, for the function $f_{max}(t) := \{n \in \mathbb{N} \mid n < m,$ for some $m \in f(t)\}$ (which is well-defined, as $f(t)$ is non-empty for $t \in T'$), it holds that $T[f_{max},\infty] \models^l \bigovee_{i\in I} \alpha_i^\psi$, which in turn is equivalent with $T[f_{max},\infty] \models^l \alpha_{k'}^\psi$ for some $k' \in I$. By downward closure the latter holds for all intermediary functions, and thus $T \models^l \alpha_{k'}^\psi \,\mathsf{U}\, \alpha_k^\theta$ and finally $T \models^l \bigovee_{i\in I, j\in J}(\alpha_i^\theta \,\mathsf{U}\, \alpha_j^\psi)$ as wanted. The converse is analogous.

For showing the size estimates stated in the theorem, it suffices to note that our translation to $\ovee$-disjunctive normal from can be equivalently stated: $\varphi \equiv \bigovee_{i\in I} \alpha_i^\psi = \bigovee_{f\in F} \varphi^f$, where F is the set of all selection functions f that select, separately for each occurrence, either the left disjunct ψ or the right disjunct θ of each subformula of the form $\psi \ovee \theta$ of φ, and φ^f denotes the formula obtained from φ by substituting each occurrence of a subformula of type $(\psi \ovee \theta)$ by $f(\psi \ovee \theta)$. The size estimates follow immediately from this observation. ◀

Proofs for the following two corollaries can be found in the full version of this article [13].

▶ **Corollary 11.** *For every flat* $\mathrm{TeamLTL}^l(\ovee)$*-formula there exists an equivalent* $\mathrm{TeamLTL}^l$*-formula.*

▶ **Corollary 12.** $\mathrm{TeamLTL}^l(\ovee) < \mathrm{TeamLTL}(\ovee)$.

A normal form, similar to the one in Theorem 10, can also be obtained for $\mathrm{TeamLTL}(\sim)$. However, since the extension is not downward closed, it only holds for a specific fragment of the logic. The following normal form has been introduced and used in [17, 16] to analyse the complexity of modal team logic and FO^2 in the team semantics context. Below φ^d denotes a formula obtained by transforming $\neg\varphi$ into negation normal form in the standard way in LTL.

▶ **Definition 13.** *A formula φ is quasi-flat if φ is of the form:* $\bigovee_{i\in I}(\alpha_i \wedge \bigwedge_{j\in J_i} \exists\beta_{i,j})$*, where α_i and $\beta_{i,j}$ are* LTL*-formulae, and $\exists\beta_{i,j}$ is an abbreviation for the formula* $\sim \beta_{i,j}^d$.

Note that, for LTL-formulae α and β, we have $T \models^l \alpha$ if and only if $t \models \alpha$, for all $t \in T$. Moreover $T \models^l \exists\beta$, if and only if there exists some trace $t \in T$ such that $t \models \beta$.

▶ **Theorem 14.** *Every left-downward closed formula* $\varphi \in \mathrm{TeamLTL}^l(\sim, \otimes)$ *is logically equivalent to a quasi-flat formula* φ^**.*

Proof. Proof by induction over the structure of φ. Atoms are flat, and hence are in the normal form. The translations and the proofs of correctness for the cases of conjunction, disjunction, and Boolean negation are analogous to the simpler modal framework of [17, 16].

Suppose $\varphi = \psi \wedge \theta$ and assume that ψ is equivalent to $\bigotimes_{i\in I}(\alpha_i^\psi \wedge \bigwedge_{j\in J_i} \exists\beta_{i,j}^\psi)$ and θ to $\bigotimes_{i\in I'}(\alpha_i^\theta \wedge \bigwedge_{j\in J_i'} \exists\beta_{i,j}^\theta)$. By the distributive laws of conjunction and disjunction, φ is clearly equivalent to

$$\bigotimes_{i\in I,k\in I'} (\alpha_i^\psi \wedge \alpha_k^\theta \wedge \bigwedge_{j\in J_i} \exists\beta_{i,j}^\psi \wedge \bigwedge_{j\in J_k'} \exists\beta_{k,j}^\theta).$$

Suppose $\varphi = \psi \vee \theta$. By the induction hypothesis and an argument analogous to the disjunction case of the proof of Theorem 10, φ is equivalent to

$$\bigotimes_{i\in I,k\in I'} \big((\alpha_i^\psi \wedge \bigwedge_{j\in J_i} \exists\beta_{i,j}^\psi) \vee (\alpha_k^\theta \wedge \bigwedge_{j\in J_k'} \exists\beta_{k,j}^\theta)\big). \tag{1}$$

The above formula expresses that T can be split into two parts: T_1 in which each trace satisfies α_i and the subformulae $\beta_{i,j}$ are satisfied by some traces, and T_2 in which each trace satisfies α_k and the subformulae $\beta_{k,j}$ are satisfied by some traces. But this is equivalent to saying that T can be split into two parts: T_1 in which each trace satisfies α_i, and T_2 in which each trace satisfies α_k; and the subformulae $\alpha_i \wedge \beta_{i,j}$ and $\alpha_k \wedge \beta_{k,j}$ are satisfied by some traces in T, and thus the formula (1) is equivalent with

$$\bigotimes_{i\in I,k\in I'} \big((\alpha_i^\psi \vee \alpha_k^\theta) \wedge \bigwedge_{j\in J_i} \exists(\alpha_i^\psi \wedge \beta_{i,j}^\psi) \wedge \bigwedge_{j\in J_k'} \exists(\alpha_j^\theta \wedge \beta_{k,j}^\theta)\big)$$

that is in the normal form.

Suppose $\varphi = {\sim}\psi$ and assume that ψ is equivalent to $\bigotimes_{i\in I}(\alpha_i \wedge \bigwedge_{j\in J_i} \exists\beta_{i,j})$. Now φ is clearly equivalent to $\bigwedge_{i\in I}(\exists\alpha_i^d \otimes \bigotimes_{j\in J_i} \beta_{i,j}^d)$. This formula can be expanded back to the normal form with exponential blow-up using the distributivity law of propositional logic.

Suppose $\varphi = \bigcirc\psi$ and assume that ψ is equivalent to $\bigotimes_{i\in I}(\alpha_i \wedge \bigwedge_{j\in J_i} \exists\beta_{i,j})$. It is now easy to check that φ is equivalent to $\bigotimes_{i\in I}(\bigcirc\alpha_i \wedge \bigwedge_{j\in J_i} \exists\bigcirc\beta_{i,j})$.

Suppose $\varphi = \mathsf{G}\,\psi$. Since φ is left-downward closed, ψ is equivalent with a formula of the form $\bigotimes_i \alpha_i$, which can be transformed to the normal form by Theorem 10.

Suppose $\varphi = \psi\,\mathsf{U}\,\theta$. By assumption φ is left-downward closed hence ψ is equivalent with a formula of the form $\bigotimes_{i\in I} \alpha_i^\psi$ (by the previous theorem) and θ is equivalent to $\bigotimes_{k\in I'}(\alpha_k^\theta \wedge \bigwedge_{j\in J_k} \exists\beta_{k,j}^\theta)$. Now using the fact that ψ is downward closed, it is easy to see that φ is logically equivalent with the formula:

$$\bigotimes_{i\in I,k\in I'} \big(\alpha_i^\psi\,\mathsf{U}(\alpha_k^\theta \wedge \bigwedge_{j\in J_k} \exists\beta_{k,j}^\psi)\big). \tag{2}$$

It now suffices to show that the disjuncts (for any $i \in I, k \in I'$) of (2) can be equivalently expressed as:

$$\big(\alpha_i^\psi\,\mathsf{U}\,\alpha_k^\theta \wedge \bigwedge_{j\in J_k} \exists(\alpha_i^\psi\,\mathsf{U}(\alpha_k^\theta \wedge \beta_{k,j}^\psi)\big). \tag{3}$$

We will show the logical implication from (3) to (2). Assume

$$T \models^l \big(\alpha_i^\psi\,\mathsf{U}\,\alpha_k^\theta \wedge \bigwedge_{j\in J_k} \exists(\alpha_i^\psi\,\mathsf{U}(\alpha_k^\theta \wedge \beta_{k,j}^\psi)\big).$$

Let f be such that $T[f,\infty] \models^l \alpha_k^\theta$ and that $T[g,\infty] \models^l \alpha_i^\psi$, for all $g < f$. In order to show

$$T \models^l \alpha_i^\psi \,\mathsf{U}(\alpha_k^\theta \wedge \bigwedge_{j\in J_k} \exists\beta_{k,j}^\psi), \tag{4}$$

we need to make sure that traces witnessing the truth of the formulae $\exists\beta_{k,j}^\psi$ can be found in $T[f,\infty]$. Here we can use the assumption that $T \models^l \bigwedge_{j\in J_k} \exists(\alpha_i^\psi \,\mathsf{U}(\alpha_k^\theta \wedge \beta_{k,j}^\psi))$ implying that for each $j \in J_k$ there exists $t_j \in T$ such that $t_j \models \alpha_i^\psi \,\mathsf{U}(\alpha_k^\theta \wedge \beta_{k,j}^\psi)$. Let now n_j be such that $t_j[n_j,\infty] \models \alpha_k^\theta \wedge \beta_{k,j}^\psi$ and that $t_j[l,\infty] \models \alpha_i^\psi$ for all $l < n_j$. Now by the flatness of the formulae $\alpha_i^\psi, \alpha_k^\theta$, and $\beta_{k,j}^\psi$, the function f' defined by

$$f'(t) := \begin{cases} f(t) \cup \{t_j[n_j,\infty]\} & \text{if } t = t_j, \text{ for some } j \in J_k \\ f(t) & \text{otherwise} \end{cases}$$

witnesses (4). The converse is proved analogously. ◂

The following example indicates that the restriction to left-downward closed formulae is necessary for the proof to work in the above theorem. An alternate proof that does not require the restriction to left-downward closed formulae may still exist.

▶ **Example 15.** Let φ be the formula $\mathsf{G}(\exists p_1 \circledvee \exists p_2)$ and $T := \{t\}$, where $t := (\{p_1\}\{p_2\})^\omega$. It is now easy to check that $T \models^l \varphi$ but $T \not\models^l \mathsf{G}\,\exists p_i$ for $i \in \{1,2\}$.

5 Computational Properties

In this section we analyse the computational properties of the logics studied in the previous section. We focus on the complexity of the model checking and satisfiability problems.

For the model checking problem one has to determine whether a team of traces generated by a given finite Kripke structure satisfies a given formula. We consider Kripke structures of the form $\mathrm{K} := (W, R, \eta, w_0)$, where W is a finite set of states, $R \subseteq W^2$ a left-total transition relation, $\eta\colon W \to 2^{\mathrm{AP}}$ a labelling function, and $w_0 \in W$ an initial state of W. A path σ through K is an infinite sequence $\sigma \in W^\omega$ such that $\sigma[0] := w_0$ and $(\sigma[i], \sigma[i+1]) \in R$ for every $i \geq 0$. The *trace of* σ is defined as $t(\sigma) := \eta(\sigma[0])\eta(\sigma[1])\cdots \in (2^{\mathrm{AP}})^\omega$. A Kripke structure K then *generates* the trace set $\mathrm{Traces}(\mathrm{K}) := \{t(\sigma) \mid \sigma \text{ is a path through K}\}$.

▶ **Definition 16.** *The* model checking problem *of a logic $\mathcal{L}$ is the following decision problem: Given a formula $\varphi \in \mathcal{L}$ and a Kripke structure K over* AP*, determine whether Traces$(K) \models \varphi$, The* (countable) satisfiability problem *of a logic $\mathcal{L}$ is the following decision problem: Given a formula $\varphi \in \mathcal{L}$, determine whether $T \models \varphi$ for some (countable) $T \neq \emptyset$.*

Below we will use the fact that the model checking and satisfiability problems of LTL are PSPACE-complete [25]. Furthermore, we use the facts that the satisfiability problem of propositional team logic PL($\sim$) is ATIME-ALT(exp, poly)-complete [9], and that the complexity of modal team logic is complete for the class $\mathrm{TOWER(poly)} := \mathrm{TIME}(\exp_{n^{O(1)}}(1))$, where $\exp_0(1) := 1$ and $\exp_{k+1}(1) := 2^{\exp_k(1)}$ [17, 16].

▶ **Theorem 17.**
1. *The model checking and satisfiability problems of* $\mathrm{TeamLTL}^l(\circledvee)$ *are* PSPACE*-complete.*
2. *The model checking and satisfiability problems of the left-flat fragment of* $\mathrm{TeamLTL}(\circledvee)$ *are* PSPACE*-complete.*

3. *The model checking problem of the left-downward closed fragment of* $\mathrm{TeamLTL}^l(\sim, \otimes)$ *is* PSPACE*-hard and it is contained in* TOWER(poly)*.*
4. *The satisfiability problem of the left-downward closed fragment of* $\mathrm{TeamLTL}^l(\sim, \otimes)$ *is* ATIME-ALT(exp, poly)*-hard and it is contained in* TOWER(poly)*.*

Proof. Let us first consider the proofs of claims 1 and 2. Note that PSPACE-hardness holds already for LTL-formulae, hence it suffices to show containment in PSPACE. Furthermore, note that 2 follows immediately from 1 and Theorem 8. Assume a formula $\varphi \in \mathrm{TeamLTL}^l(\otimes)$ and a Kripke structure K is given as input. By Theorem 10, φ is logically equivalent with a formula of the form $\bigotimes_{f \in F} \varphi^f$, where f varies over selection functions selecting, separately for each occurrence, either the left disjunct ψ or the right disjunct θ of each subformula of the form $\psi \otimes \theta$ of φ. Now, without constructing the full formula $\bigotimes_{f \in F} \varphi^f$, using polynomial space with respect to the size of φ it is possible to check whether $\mathrm{Traces(K)} \models \varphi_f$ for some $f \in F$. Hence the upper bound follows from the fact that LTL model checking is in PSPACE. The upper bound for satisfiability follows analogously.

Let us then consider the proof of claim (4). The proof of claim (3) is analogous. For the lower bound it suffices to note that propositional team logic $\mathrm{PL}(\sim)$ is a fragment of the left-downward closed fragment of $\mathrm{TeamLTL}^l(\sim, \otimes)$ and hence its satisfiability problem can be trivially reduced to the satisfiability problem of the left-downward closed fragment. Therefore ATIME-ALT(exp, poly)-hardness follows by the result of [9].

For the upper bound we first transform an input formula φ into an equivalent quasi-flat formula of the form $\bigotimes_{i \in I}(\alpha_i \wedge \bigwedge_{j \in J_i} \exists \beta_{i,j})$. Analogously to [17, 16], this formula can be computed in time $\mathrm{TIME}(\exp_{O(|\varphi|)}(1))$. It is now easy to see that the quasi-flat formula is satisfiable iff there exists $i \in I$, such that $\mathrm{SAT}(\alpha_i \wedge \beta_{i,j}) = 1$ for all $j \in J_i$. Since LTL-satisfiability checking is contained in $\mathrm{PSPACE} \subseteq \mathrm{TIME}(2^{n^{O(1)}})$, the overall complexity of the above procedure is in $\mathrm{TIME}(\exp_{(|\varphi|^{O(1)})}(1))$. ◀

6 Conclusion

We introduced a novel set-based semantics for asynchronous TeamLTL. We showed several results on the expressive power and complexity of the extensions of $\mathrm{TeamLTL}^l$ by the Boolean disjunction $\mathrm{TeamLTL}^l(\otimes)$ and by the Boolean negation $\mathrm{TeamLTL}^l(\sim)$. In particular, our results show that the complexity properties of the former logic are comparable to that of LTL and that the left-downward closed fragment of the latter has also decidable model-checking and satisfiability problems. See Table 1 on page 4 for an overview of our expressivity results and Table 2 for our complexity results. We obtained these results on $\mathrm{TeamLTL}^l(\otimes)$ and $\mathrm{TeamLTL}^l(\sim)$ via normal forms that also allowed us to relate the expressive power of these logics to the corresponding logics in the strict semantics. Our results show that, while the synchronous TeamLTL can be viewed as a fragment of second-order logic, the asynchronous $\mathrm{TeamLTL}(\otimes)$ under the lax semantics is a sublogic of HyperLTL (see [2] for a definition). Furthermore, our decidability results show, e.g, that it will probably be possible to devise a complete proof system for the logic. The full version of this article [13] relates and applies our results to recently defined logics whose asynchronicity is formalised via time evaluation functions [7]. We conclude with open questions:

- Does Theorem 14 extend to all formulae of $\mathrm{TeamLTL}^l(\sim)$? Note that any quasi-flat–$\mathrm{TeamLTL}(\sim)$-formula can be rewritten in HyperLTL.
- Can the result (iii) of Theorem 17 be accompanied by an matching lower bound (i.e., TOWER(poly)-hardness result)?

- Can a syntactic characterisation (similar to Corollary 11) be obtained for the downward closed fragment of $\text{TeamLTL}^l(\sim)$? We believe that $\text{TeamLTL}^l(\otimes)$ is a promising candidate, as its extensions with infinite conjunctions and disjunctions suffices for all downward closed properties of teams.
- What is the complexity of model checking for $\text{TeamLTL}(\otimes)$ under the strict semantics?

References

1. Jan Baumeister, Norine Coenen, Borzoo Bonakdarpour, Bernd Finkbeiner, and César Sánchez. A temporal logic for asynchronous hyperproperties. In *CAV (1)*, volume 12759 of *Lecture Notes in Computer Science*, pages 694–717. Springer, 2021.
2. Michael R. Clarkson, Bernd Finkbeiner, Masoud Koleini, Kristopher K. Micinski, Markus N. Rabe, and César Sánchez. Temporal logics for hyperproperties. In *POST 2014*, pages 265–284, 2014.
3. Michael R. Clarkson and Fred B. Schneider. Hyperproperties. *Journal of Computer Security*, 18(6):1157–1210, 2010.
4. Norine Coenen, Bernd Finkbeiner, Christopher Hahn, and Jana Hofmann. The hierarchy of hyperlogics. In *LICS 2019*, pages 1–13. IEEE, 2019.
5. Bernd Finkbeiner, Christopher Hahn, Philip Lukert, Marvin Stenger, and Leander Tentrup. Synthesis from hyperproperties. *Acta Informatica*, 57(1-2):137–163, 2020. `doi:10.1007/s00236-019-00358-2`.
6. Pietro Galliani. Inclusion and exclusion dependencies in team semantics: On some logics of imperfect information. *Annals of Pure and Applied Logic*, 163(1):68–84, 2012.
7. Jens Oliver Gutsfeld, Arne Meier, Christoph Ohrem, and Jonni Virtema. Temporal team semantics revisited. In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2–5, 2022*, pages 44:1–44:13. ACM, 2022. `doi:10.1145/3531130.3533360`.
8. Jens Oliver Gutsfeld, Markus Müller-Olm, and Christoph Ohrem. Automata and fixpoints for asynchronous hyperproperties. *Proc. ACM Program. Lang.*, 5(POPL):1–29, 2021. `doi:10.1145/3434319`.
9. Miika Hannula, Juha Kontinen, Jonni Virtema, and Heribert Vollmer. Complexity of propositional logics in team semantic. *ACM Trans. Comput. Log.*, 19(1):2:1–2:14, 2018.
10. Lauri Hella, Antti Kuusisto, Arne Meier, and Jonni Virtema. Model checking and validity in propositional and modal inclusion logics. *J. Log. Comput.*, 29(5):605–630, 2019. `doi:10.1093/logcom/exz008`.
11. Juha Kontinen, Julian-Steffen Müller, Henning Schnoor, and Heribert Vollmer. Modal independence logic. In Rajeev Goré, Barteld P. Kooi, and Agi Kurucz, editors, *Advances in Modal Logic 10, invited and contributed papers from the tenth conference on "Advances in Modal Logic," held in Groningen, The Netherlands, August 5-8, 2014*, pages 353–372. College Publications, 2014. URL: `http://www.aiml.net/volumes/volume10/Kontinen-Mueller-Schnoor-Vollmer.pdf`.
12. Juha Kontinen and Max Sandström. On the expressive power of TeamLTL and first-order team logic over hyperproperties. In *WoLLIC*, volume 13038 of *Lecture Notes in Computer Science*, pages 302–318. Springer, 2021.
13. Juha Kontinen, Max Sandström, and Jonni Virtema. Set semantics for asynchronous TeamLTL: Expressivity and complexity. *CoRR*, abs/2304.10915, 2023. `doi:10.48550/arXiv.2304.10915`.
14. Andreas Krebs, Arne Meier, and Jonni Virtema. A team based variant of CTL. In Fabio Grandi, Martin Lange, and Alessio Lomuscio, editors, *22nd International Symposium on Temporal Representation and Reasoning, TIME 2015, Kassel, Germany, September 23-25, 2015*, pages 140–149. IEEE Computer Society, 2015. `doi:10.1109/TIME.2015.11`.
15. Andreas Krebs, Arne Meier, Jonni Virtema, and Martin Zimmermann. Team Semantics for the Specification and Verification of Hyperproperties. In Igor Potapov, Paul Spirakis, and James Worrell, editors, *MFCS 2018*, volume 117, pages 10:1–10:16, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

16 Martin Lück. Axiomatizations of team logics. *Ann. Pure Appl. Log.*, 169(9):928–969, 2018.

17 Martin Lück. On the complexity of team logic and its two-variable fragment. In *MFCS*, volume 117 of *LIPIcs*, pages 27:1–27:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

18 Martin Lück. On the complexity of linear temporal logic with team semantics. *Theoretical Computer Science*, 2020.

19 Martin Lück. *Team logic: axioms, expressiveness, complexity.* PhD thesis, University of Hanover, Hannover, Germany, 2020. URL: https://www.repo.uni-hannover.de/handle/123456789/9430.

20 John McLean. Proving noninterference and functional correctness using traces. *Journal of Computer Security*, 1(1):37–58, 1992. doi:10.3233/JCS-1992-1103.

21 Nir Piterman and Amir Pnueli. Temporal logic and fair discrete systems. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 27–73. Springer, 2018. doi:10.1007/978-3-319-10575-8_2.

22 Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science*, pages 46–57. IEEE Computer Society, 1977.

23 Markus N. Rabe. *A Temporal Logic Approach to Information-Flow Control.* PhD thesis, Saarland University, 2016.

24 A. W. Roscoe. CSP and determinism in security modelling. In *Proceedings of the 1995 IEEE Symposium on Security and Privacy, Oakland, California, USA, May 8-10, 1995*, pages 114–127. IEEE Computer Society, 1995. doi:10.1109/SECPRI.1995.398927.

25 A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *J. ACM*, 32(3):733–749, July 1985. doi:10.1145/3828.3837.

26 Jouko Väänänen. *Dependence Logic.* Cambridge University Press, 2007.

27 Jonni Virtema. Complexity of validity for propositional dependence logics. *Inf. Comput.*, 253:224–236, 2017. doi:10.1016/j.ic.2016.07.008.

28 Jonni Virtema, Jana Hofmann, Bernd Finkbeiner, Juha Kontinen, and Fan Yang. Linear-time temporal logic with team semantics: Expressivity and complexity. *CoRR*, abs/2010.03311, 2020. arXiv:2010.03311.

29 Jonni Virtema, Jana Hofmann, Bernd Finkbeiner, Juha Kontinen, and Fan Yang. Linear-time temporal logic with team semantics: Expressivity and complexity. In Mikolaj Bojanczyk and Chandra Chekuri, editors, *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2021, December 15-17, 2021, Virtual Conference*, volume 213 of *LIPIcs*, pages 52:1–52:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.FSTTCS.2021.52.

30 Steve Zdancewic and Andrew C. Myers. Observational determinism for concurrent program security. In *16th IEEE Computer Security Foundations Workshop (CSFW-16 2003), 30 June – 2 July 2003, Pacific Grove, CA, USA*, page 29. IEEE Computer Society, 2003. doi:10.1109/CSFW.2003.1212703.