New developments in temporal team semantic

Jonni Virtema

University of Sheffield, UK

Logic, databases and complexity: new methods and challenges
17 February 2022

Main references

- Team Semantics for the Specification and Verification of Hyperproperties.
  MFCS 2018 paper with A. Krebs, A. Meier, and M. Zimmermann.
- Linear-time Temporal Logic with Team Semantics: Expressivity and Complexity.
  FSTTCS 2021 paper with B. Finkbeiner, J. Hofmann, J. Kontinen, and F. Yang.
- Temporal Team Semantics Revisited.
  Submitted manuscript with J. Gutsfeld, A. Meier, and C. Ohrem.

Logics for verification and specification of concurrent systems

Basic setting:

- System (e.g., piece of software or hardware)
  $\rightsquigarrow$ Kripke structure depicting the behaviour of the system
- A single run of the system
  $\rightsquigarrow$ a trace generated by the Kripke structure
- A property of the system (e.g., every request is eventually granted)
  $\rightsquigarrow$ a formula of some formal language expressing the property.

Model checking:

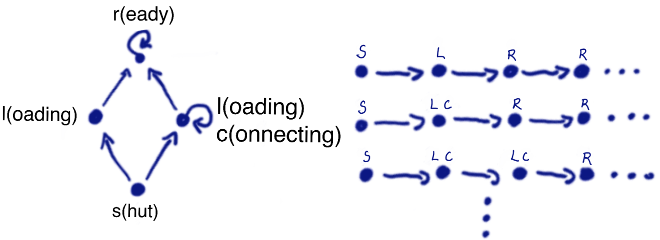
- Check whether a given system satisfies a given specification.

SAT solving:

- Check whether a given specification (or collection of) can be realised.

Traceproperties and hyperproperties

Opening your office computer after holidays:



Traceproperties hold in a system if each trace (in isolation) has the property:

- The computer will be eventually ready (or will be loading forever).

Hyperproperties are properties of sets of traces:

- The computer will be ready in bounded time.

Quantifier extensions vs. team semantics

Classical setting:

- LTL, QPTL, CTL, etc. vs. HyperLTL, HyperQPTL, HyperCTL, etc. are prominent logics for traceproperties vs. hyperproperties of systems
  - Traceproperty: Each request is eventually granted (properties of traces)
  - Hyperproperty: Each request is granted in bounded time (properties of sets of traces)
- HyperLogics are of high complexity or undecidable.
  Not well suited for properties involving unbounded number of traces.

Alternative way by using team semantics

- Temporal logics with team semantics for expressing hyperproperties
  Purely modal logic & well suited for properties of unbounded number of traces.
- Expressivity: How TeamLTL variants relate to HyperLogics?
- Complexity: Where is the undecidability frontier of TeamLTL extensions?
  - A large EXPTIME fragment: left-flat and downward closed logics
  - Already TeamLTL with inclusion atoms and Boolean disjunctions is undecidable

Logics for traceproperties and hyperproperties

- Linear-time temporal logic (LTL) is one of the most prominent logics for the specification and verification of reactive and concurrent systems.
- Model checking tools like SPIN and NuSMV automatically verify whether a given computer system is correct with respect to its LTL specification.
- One reason for the success of LTL over first-order logic is that LTL is a purely modal logic and thus has many desirable properties.
  - LTL is decidable (PSPACE-complete model checking and satisfiability).
  - $\mathrm{FO}^2(\leq)$ and $\mathrm{FO}^3(\leq)$ SAT are NEXPTIME-complete and non-elementary.
- Caveat: LTL can specify only traceproperties.

Logics for traceproperties and hyperproperties

Recipe for logics for hyperproperties:
A logic for traceproperties $\rightsquigarrow$ add trace quantifiers

In LTL the satisfying object is a trace: $T \models \varphi$ iff $\forall t \in T : t \models \varphi$

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid X\varphi \mid \varphi U \varphi$$

In HyperLTL the satisfying object is a set of traces and a trace assignment: $\Pi \models_T \varphi$

$$\varphi ::= \exists \pi \varphi \mid \forall \pi \varphi \mid \psi$$
$$\psi ::= p_\pi \mid \neg\psi \mid (\psi \vee \psi) \mid X\psi \mid \psi U \psi$$

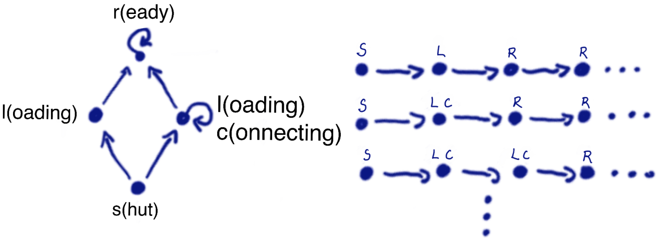
HyperQPTL extends HyperLTL by (uniform) quantification of propositions: $\exists p \varphi$, $\forall p \varphi$

Logics for traceproperties and hyperproperties

- Quantification based logics for hyperproperties: HyperLTL, HyperCTL, etc.
- Retain some desirable properties of LTL, but are not purely modal logics
  - Model checking for $\exists^*$HyperLTL and HyperLTL are PSPACE and non-elementary.
  - HyperLTL satisfiability is highly undecidable.
  - HyperLTL formulae express properties expressible using fixed finite number of traces.

Logics for traceproperties and hyperproperties

- Quantification based logics for hyperproperties: HyperLTL, HyperCTL, etc.
- Retain some desirable properties of LTL, but are not purely modal logics
  - Model checking for $\exists^*$HyperLTL and HyperLTL are PSPACE and non-elementary.
  - HyperLTL satisfiability is highly undecidable.
  - HyperLTL formulae express properties expressible using fixed finite number of traces.
- Bounded termination is not definable in HyperLTL (but is in HyperQPTL)

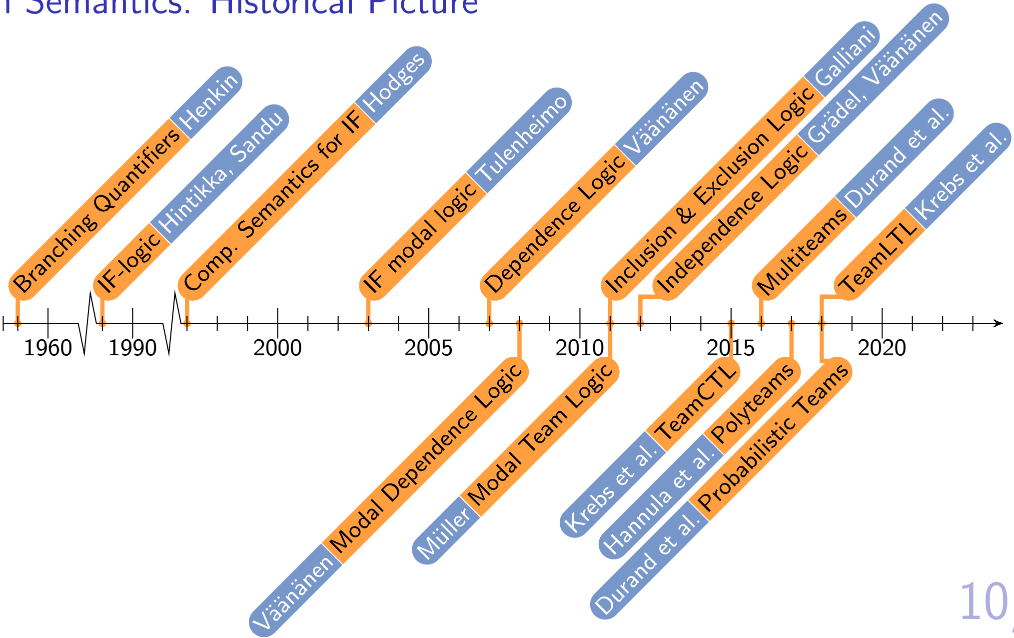


- Team semantics is a candidate for a purely modal logic without the above caveat.

Core of Team Semantics

- In most studied logics formulae are evaluated in a single state of affairs.
  E.g.,
  - a first-order assignment in first-order logic,
  - a propositional assignment in propositional logic,
  - a possible world of a Kripke structure in modal logic.
- In team semantics sets of states of affairs are considered.
  E.g.,
  - a set of first-order assignments in first-order logic,
  - a set of propositional assignments in propositional logic,
  - a set of possible worlds of a Kripke structure in modal logic.
- These sets of things are called teams.

Team Semantics: Historical Picture



LTL, HyperLTL, and TeamLTL

In LTL the satisfying object is a trace: $T \models \varphi$ iff $\forall t \in T : t \models \varphi$

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid X\varphi \mid \varphi U \varphi$$

In HyperLTL the satisfying object is a set of traces and a trace assignment: $\Pi \models_T \varphi$

$$\varphi ::= \exists\pi\varphi \mid \forall\pi\varphi \mid \psi$$
$$\psi ::= p_\pi \mid \neg\psi \mid (\psi \vee \psi) \mid X\psi \mid \psi U \psi$$

In TeamLTL the satisfying object is a set of traces. We use team semantics: $(T, i) \models \varphi$

$$\varphi ::= p \mid \neg p \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid X\varphi \mid \varphi U \mid \varphi W \varphi$$

\+ new atomic statements (dependence and inclusion atoms: $\mathrm{dep}(\vec{p}, q)$, $\vec{p} \subseteq \vec{q}$)
\+ additional connectives (Boolean disjunction, contradictory negation, etc.)

Extensions are a well-defined way to delineate expressivity and complexity

Examples: HyperLTL vs. TeamLTL

Temporal team semantics is universal and synchronous

$$(T, i) \models p \text{ iff } \forall t \in T : t[i](p) = 1 \qquad (T, i) \models \neg p \text{ iff } \forall t \in T : t[i](p) = 0$$

$$(T, i) \models \mathrm{F}\varphi \text{ iff } (T, j) \models \varphi \text{ for some } j \geq i \quad (T, i) \models \mathrm{G}\varphi \text{ iff } (T, j) \models \varphi \text{ for all } j \geq i$$

Examples: HyperLTL vs. TeamLTL

Temporal team semantics is universal and synchronous

$$(T,i) \models p \text{ iff } \forall t \in T : t[i](p) = 1 \qquad (T,i) \models \neg p \text{ iff } \forall t \in T : t[i](p) = 0$$

$$(T,i) \models \mathsf{F}\varphi \text{ iff } (T,j) \models \varphi \text{ for some } j \geq i \quad (T,i) \models \mathsf{G}\varphi \text{ iff } (T,j) \models \varphi \text{ for all } j \geq i$$

There is a timepoint (common for all traces) after which a does not occur.
Not expressible in HyperLTL, but is in HyperQPTL.

$$\exists p \forall \pi \, \mathsf{F} p \wedge \mathsf{G}(p \rightarrow \mathsf{G} \neg a_\pi)$$

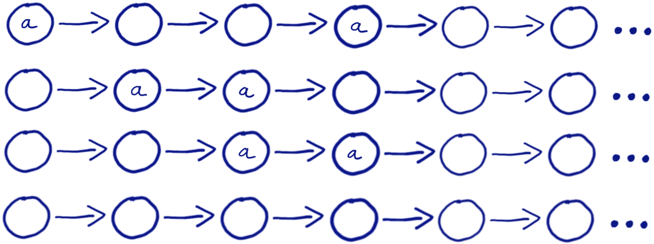
Expressible in synchronous TeamLTL: $\mathsf{FG} \neg a$

Examples: HyperLTL vs. TeamLTL

There is a timepoint (common for all traces) after which a does not occur.
Not expressible in HyperLTL, but is in HyperQPTL.

$$\exists p \forall \pi \, \mathsf{F} p \wedge \mathsf{G}(p \rightarrow \mathsf{G} \neg a_\pi)$$

Expressible in synchronous TeamLTL: $\mathsf{FG}\,\neg a$

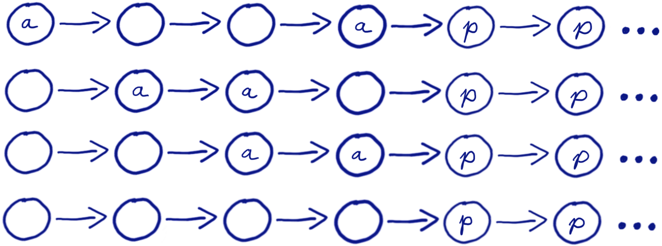


Examples: HyperLTL vs. TeamLTL

There is a timepoint (common for all traces) after which a does not occur.
Not expressible in HyperLTL, but is in HyperQPTL.

$$\exists p \, \forall \pi \, \mathsf{F} p \wedge \mathsf{G}(p \rightarrow \mathsf{G} \neg a_\pi)$$

Expressible in synchronous TeamLTL: $\mathsf{FG} \neg a$

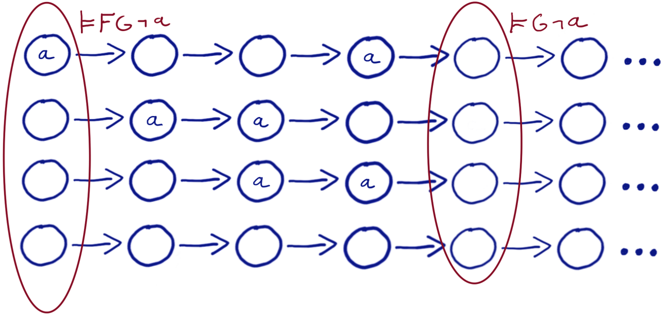


Examples: HyperLTL vs. TeamLTL

There is a timepoint (common for all traces) after which a does not occur.
Not expressible in HyperLTL, but is in HyperQPTL.

$$\exists p \forall \pi \, \mathsf{F} p \wedge \mathsf{G}(p \rightarrow \mathsf{G} \neg a_\pi)$$

Expressible in synchronous TeamLTL: $\mathsf{FG} \neg a$



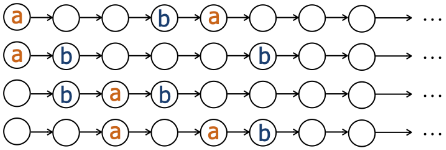
Examples: HyperLTL vs. TeamLTL

A trace-set T satisfies $\varphi \vee \psi$ if it decomposed to sets T_φ and T_ψ satisfying φ and ψ.

$$(T, i) \models \varphi \vee \psi \text{ iff } (T_1, i) \models \varphi \text{ and } (T_2, i) \models \psi, \text{ for some } T_1 \cup T_2 = T$$
$$(T, i) \models \varphi \wedge \psi \text{ iff } (T, i) \models \varphi \text{ and } (T, i) \models \psi$$

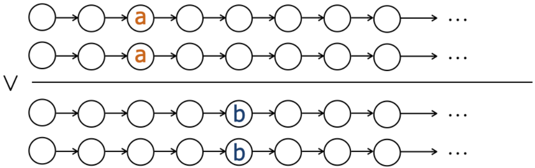
HyperLTL:

$\forall \pi . \forall \pi' . \ \mathrm{F}((a_\pi \wedge a_{\pi'}) \vee (b_\pi \wedge b_{\pi'}))$



TeamLTL:

$(\mathrm{F}\ a) \vee (\mathrm{F}\ b)$



Examples: HyperLTL vs. TeamLTL

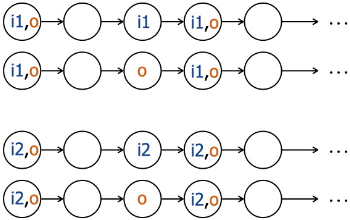
Dependence atom $\text{dep}(p_1, \ldots, p_m, q)$ states that $p_1, \ldots, p_m$ functionally determine q:

$$(T, i) \models \text{dep}(p_1, \ldots, p_m, q) \text{ iff } \forall t, t' \in T\Big(\bigwedge_{1 \le j \le m} t[i](p_j) = t'[i](p_j)\Big) \Rightarrow (t[i](q) = t'[i](q))$$

TeamLTL:

$$(\mathrm{G}\ dep(i1, o)) \vee (\mathrm{G}\ dep(i2, o))$$

Nondeterministic dependence: "*o* either depends on i1 or on i2"



"whenever the traces agree on i1, they agree on o"

$\vee$

"whenever the traces agree on i2, they agree on o"

Temporal team semantics

Definition

Temporal team is (T, i), where T a set of traces and $i \in \mathbb{N}$.

$$
\begin{array}{lll}
(T, i) \models p & \text{iff} & \forall t \in T : t[0](p) = 1 \\
(T, i) \models \neg p & \text{iff} & \forall t \in T : t[0](p) = 0 \\
(T, i) \models \phi \wedge \psi & \text{iff} & (T, i) \models \phi \text{ and } (T, i) \models \psi \\
(T, i) \models \phi \vee \psi & \text{iff} & (T_1, i) \models \phi \text{ and } (T_2, i) \models \psi, \text{ for some } T_1, T_2 \text{ s.t. } T_1 \cup T_2 = T \\
(T, i) \models \mathrm{X}\varphi & \text{iff} & (T, i+1) \models \varphi \\
(T, i) \models \phi \mathrm{U} \psi & \text{iff} & \exists k \geq i \text{ s.t. } (T, k) \models \psi \text{ and } \forall m : i \leq m < k \Rightarrow (T, m) \models \phi \\
(T, i) \models \phi \mathrm{W} \psi & \text{iff} & \forall k \geq i : (T, k) \models \phi \text{ or } \exists m \text{ s.t. } i \leq m \leq k \text{ and } (T, m) \models \psi
\end{array}
$$

As usual $\mathrm{F}\varphi := (\top \mathrm{U} \varphi)$ and $\mathrm{G}\varphi := (\varphi \mathrm{W} \bot)$.

$\mathrm{TeamLTL}(\oslash, \subseteq)$ is the extension with the atoms and extra connectives in the brackets.

Generalised atoms and complete logics

Let B be a set of n-ary Boolean relations. We define the property $[\varphi_1, \ldots, \varphi_n]_B$ for an n-tuple $(\varphi_1, \ldots, \varphi_n)$ of LTL-formulae:

$$(T, i) \models [\varphi_1, \ldots, \varphi_n]_B \quad \text{iff} \quad \{(\llbracket\phi_1\rrbracket_{(t,i)}, \ldots, \llbracket\phi_n\rrbracket_{(t,i)}) \mid t \in T\} \in B.$$

Generalised atoms and complete logics

Let B be a set of n-ary Boolean relations. We define the property $[\varphi_1, \ldots, \varphi_n]_B$ for an n-tuple $(\varphi_1, \ldots, \varphi_n)$ of LTL-formulae:

$$(T, i) \models [\varphi_1, \ldots, \varphi_n]_B \quad \text{iff} \quad \{(\llbracket \phi_1 \rrbracket_{(t,i)}, \ldots, \llbracket \phi_n \rrbracket_{(t,i)}) \mid t \in T\} \in B.$$

Theorem

$\mathrm{TeamLTL}(\otimes, \mathrm{NE}, \overset{1}{\mathrm{A}})$ *can express all* $[\varphi_1, \ldots, \varphi_n]_B$.

$\mathrm{TeamLTL}(\otimes, \overset{1}{\mathrm{A}})$ *can express all* $[\varphi_1, \ldots, \varphi_n]_B$, *for downward closed* B.

- B is downdard closed if $S_1 \in B$ & $S_2 \subseteq S_1$ imply $S_2 \in B$.
- $(T, i) \models \varphi \otimes \psi$ iff $(T, i) \models \varphi$ or $(T, i) \models \psi$
- $(T, i) \models \mathrm{NE}$ iff $T \neq \emptyset$.
- $(T, i) \models \mathrm{A}\varphi$ iff $(T', i) \models \varphi$, for all $T' \subseteq T$.
- $(T, i) \models \overset{1}{\mathrm{A}}\varphi$ iff $(\{t\}, i) \models \varphi$, for all $t \in T$.

Complexity results

Logic	Model Checking Result
$\mathrm{TeamLTL}$ without $\vee$	in PSPACE
k-coherent $\mathrm{TeamLTL}(\sim)$	in EXPSPACE
left-flat $\mathrm{TeamLTL}(\otimes, \overset{1}{\mathsf{A}})$	in EXPSPACE
$\mathrm{TeamLTL}(\subseteq, \otimes)$	Σ^0_1-hard
$\mathrm{TeamLTL}(\subseteq, \otimes, \mathsf{A})$	Σ^1_1-hard
$\mathrm{TeamLTL}(\sim)$	complete for third-order arithmetic [Luck 2020]

Table: Complexity results.

- k-coherence: $(T, i) \models \varphi$ iff $(S, i) \models \varphi$ for all $S \subseteq T$ s.t. $|S| \leq k$
- left-flatness: Restrict U and W syntactically to $(\overset{1}{\mathsf{A}}\varphi \mathsf{U} \psi)$ and $(\overset{1}{\mathsf{A}}\varphi \mathsf{W} \psi)$
- $\sim$ is contradictory negation and $\mathrm{TeamLTL}(\sim)$ subsumes all the other logics

Source of inclusion results

$$\mathrm{TeamLTL}(\varovee, \overset{1}{\mathsf{A}}) \leq \overset{u}{\exists}{}^{*}_{q}\forall_\pi \mathrm{HyperQPTL} \text{ (assuming left-flatness)}$$

$$\leq \exists_p \overset{u}{Q}{}^{*}_{p}\forall_\pi \mathrm{HyperQPTL}^{+} \text{ (general case)}$$

$$\wedge^{\dagger}$$

$$\mathrm{TeamLTL}(\varovee, \mathrm{NE}, \overset{1}{\mathsf{A}}) \leq \exists_p \overset{u}{Q}{}^{*}_{p}\exists^{*}_{\pi}\forall_\pi \mathrm{HyperQPTL}^{+}$$

$$|\wedge \quad \text{[Luck 2020]} \qquad \text{(assuming } k\text{-coherence)}$$

$$\mathrm{TeamLTL}(\sim) \leq \forall^{k}\mathrm{HyperLTL}$$

Table: Expressivity results. † holds since $\mathrm{TeamLTL}(\overset{1}{\mathsf{A}}, \varovee)$ is downward closed.

- $\exists_p$ is a quantification of a new proposition
- $\overset{u}{Q}{}^{*}_{p}$ is quantification of new uniform propositions (unique value for each time step)
- $\forall_\pi$ is a quantification of a trace variable

Source of Undecidability

Definition

A non-deterministic 3-counter machine M consists of a list I of n instructions that manipulate three counters C_l, C_m and C_r. All instructions are of the following forms:

- C_a^+ goto $\{j_1, j_2\}$, C_a^- goto $\{j_1, j_2\}$, if $C_a = 0$ goto $j_1$else goto j_2,

where $a \in \{l, m, r\}$, $0 \leq j_1, j_2 < n$.

- configuration: tuple (i, j, k, l), where $0 \leq i < n$ is the next instruction to be executed, and $j, k, l \in \mathbb{N}$ are the current values of the counters C_l, C_m and C_r.
- computation: infinite sequence of consecutive configurations starting from the initial configuration $(0, 0, 0, 0)$.
- computation b-recurring if the instruction labelled b occurs infinitely often in it.
- computation is lossy if the counter values can non-deterministically decrease

Theorem (Alur & Henzinger 1994, Schnoebelen 2010)

Deciding whether a given non-deterministic 3-counter machine has a (lossy) b-recurring computation for a given b is (Σ_1^0-complete) Σ_1^1-complete.

Undecidability results

Theorem

Model checking for $\mathrm{TeamLTL}(\varovee, \subseteq)$ *is* Σ^1_0*-hard.*
Model checking for $\mathrm{TeamLTL}(\varovee, \subseteq, \mathsf{A})$ *is* Σ^1_1*-hard.*

Proof Idea:

- reduce existence of *b*-recurring computation of given 3-counter machine M and instruction label b to model checking problem of $\mathrm{TeamLTL}(\varovee, \subseteq, \mathsf{A})$
- $\mathrm{TeamLTL}(\varovee, \subseteq)$ suffices to enforce lossy computation
- $(T[i, \infty], 0)$ encodes the value of counters of the ith configuration
  the value of C_a is the cardinality of the set $\{t \in T[i, \infty] \mid t[0](c_a) = 1\}$

Modes of asynchronicity

- Synchronous TeamLTL:
  - $(T, i) \models \varphi$
  - Collection of traces T with one global clock i.
- Asynchronous TeamLTL:
  - $(T, f) \models \varphi$
  - Collection of traces T with a collection of local clocks $f\colon T \to \mathbb{N}$.
  - Local clocks are completely independent.

Modes of asynchronicity

- Synchronous TeamLTL:
  - $(T, i) \models \varphi$
  - Collection of traces T with one global clock i.
- Asynchronous TeamLTL:
  - $(T, f) \models \varphi$
  - Collection of traces T with a collection of local clocks $f\colon T \to \mathbb{N}$.
  - Local clocks are completely independent.
- TeamLTL with time evaluation functions (tefs):
  - $(T, \tau) \models \varphi$
  - Collection of traces T and a tef $\tau\colon \mathbb{N} \times T \to \mathbb{N}$ relating a global clock to local clocks.
  - The behaviour of local clocks is determined by a tef.
  - Synchronous TeamLTL is an instance, where the tef is synchronous!
  - (cf. trajectories of Bonakdarpour, Prabhakar, Sánchez, NASA Formal Methods 2020)

Properties of tefs

Property	**Definition**
Monotonicity	$\forall i \in \mathbb{N} : \tau(i) \leq \tau(i+1)$
Strict Monotonicity	$\forall i \in \mathbb{N} : \tau(i) < \tau(i+1)$
Stepwiseness	$\forall i \in \mathbb{N} : \tau(i) \leq \tau(i+1) \leq \tau(i) + \vec{1}$
*Fairness	$\forall i \in \mathbb{N}\, \forall t \in T\, \exists j \in \mathbb{N} : \tau(j,t) \geq i$
*Non-Parallelism	$\forall i \in \mathbb{N} : i = \sum_{t \in T} \tau(i,t)$
*Synchronicity	$\forall i, i' \in \mathbb{N}\, \forall t \in T : \tau(i,t) = \tau(i,t')$

Table: * are optional. $\tau(i)$ is the tuple $\big(\tau(i,t)\big)_{t \in T}$ of values of local clocks at time i.

Properties of tefs

Property	**Definition**
Monotonicity	$\forall i \in \mathbb{N} : \tau(i) \leq \tau(i+1)$
Strict Monotonicity	$\forall i \in \mathbb{N} : \tau(i) < \tau(i+1)$
Stepwiseness	$\forall i \in \mathbb{N} : \tau(i) \leq \tau(i+1) \leq \tau(i) + \vec{1}$
*Fairness	$\forall i \in \mathbb{N}\, \forall t \in T\, \exists j \in \mathbb{N} : \tau(j,t) \geq i$
*Non-Parallelism	$\forall i \in \mathbb{N} : i = \sum_{t \in T} \tau(i,t)$
*Synchronicity	$\forall i, i' \in \mathbb{N}\, \forall t \in T : \tau(i,t) = \tau(i,t')$

Table: * are optional. $\tau(i)$ is the tuple $\left(\tau(i,t)\right)_{t \in T}$ of values of local clocks at time i.

- stuttering tef satisfies monotonicity
- tef satisfies strict monotonicity and stepwiseness
- synchronous tef satisfies strict monotonicity, stepwiseness, and synchronicity
- tef is initial, if $\tau(0,t) = 0$ for each $t \in T$.
- k-shifted tef if defined by $\tau[k,\infty](i,t) := \tau(i+k,t)$, for all $t \in T$, $i \in \mathbb{N}$.

Team semantics with tefs

A temporal team is a pair (T, τ), where T is a multiset of traces and τ is a tef for T.
A pair (T, τ) is called a stuttering temporal team if τ is a stuttering tef for T.

$$
\begin{array}{lll}
(T,\tau) \models p & \text{iff} & \forall t \in T : p \in t[\tau(0,t)] \\
(T,\tau) \models \neg p & \text{iff} & \forall t \in T : p \notin t[\tau(0,t)] \\
(T,\tau) \models (\varphi \wedge \psi) & \text{iff} & (T,\tau) \models \varphi \text{ and } (T,\tau) \models \psi \\
(T,\tau) \models (\varphi \vee \psi) & \text{iff} & \exists T_1 \uplus T_2 = T : (T_1,\tau) \models \varphi \text{ and } (T_2,\tau) \models \psi \\
(T,\tau) \models \mathsf{X}\varphi & \text{iff} & (T,\tau[1,\infty]) \models \varphi \\
(T,\tau) \models [\varphi \mathsf{U} \psi] & \text{iff} & \exists k \in \mathbb{N} \text{ such that } (T,\tau[k,\infty]) \models \psi \text{ and} \\
 & & \quad \forall m : 0 \leq m < k \Rightarrow (T,\tau[m,\infty]) \models \varphi \\
(T,\tau) \models [\varphi \mathsf{W} \psi] & \text{iff} & \forall k \in \mathbb{N} : (T,\tau[k,\infty]) \models \varphi \text{ or} \\
 & & \quad \exists m \text{ s.t. } m \leq k \text{ and } (T,\tau[m,\infty]) \models \psi
\end{array}
$$

Variants of TeamLTL

∃TeamLTL

- $T \models_{\exists} \varphi$ if $(T, \tau) \models \varphi$ for some initial tef of T.

∀TeamLTL

- $T \models_{\forall} \varphi$ if $(T, \tau) \models \varphi$ for all initial tefs of T.

Synchronous TeamLTL

- $T \models_{s} \varphi$ if $(T, \tau) \models \varphi$ for the unique initial synchronous tef of T.

Variants of TeamLTL

$\exists$TeamLTL

- $T \models_{\exists} \varphi$ if $(T, \tau) \models \varphi$ for some initial tef of T.

$\forall$TeamLTL

- $T \models_{\forall} \varphi$ if $(T, \tau) \models \varphi$ for all initial tefs of T.

Synchronous TeamLTL

- $T \models_{s} \varphi$ if $(T, \tau) \models \varphi$ for the unique initial synchronous tef of T.

Theorem

A formula is satisfiable in $\exists\mathrm{TeamLTL}$ *iff it is satisfiable in synchronous* $\mathrm{TeamLTL}$*.*
A formula is valid in $\forall\mathrm{TeamLTL}$ *iff it is valid in synchronous* $\mathrm{TeamLTL}$*.*

Theorem

Model checking of synchronous $\mathrm{TeamLTL}$ *reduces in linear time to the model checking of* $\exists\mathrm{TeamLTL}$ *and* $\forall\mathrm{TeamLTL}(\otimes,\mathrm{NE})$*.*

Quantifier extensions of TeamLTL

- $\mathrm{TeamCTL}^*$ has the same syntax as CTL*:

$$\varphi ::= p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \varphi \mathsf{W} \varphi \mid \exists \varphi \mid \forall \varphi$$

  The quantifiers $\exists$ and $\forall$ range over tefs:

$$(T, \tau) \models \exists \varphi \text{ iff } (T, \tau') \models \varphi \text{ for some tef } \tau' \text{ of } T \text{ s.t. } \tau'(0) = \tau(0),$$
$$(T, \tau) \models \forall \varphi \text{ iff } (T, \tau') \models \varphi \text{ for all tefs } \tau' \text{ of } T \text{ s.t. } \tau'(0) = \tau(0).$$

Quantifier extensions of TeamLTL

- $\mathrm{TeamCTL}^*$ has the same syntax as CTL*:

$$\varphi ::= p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \varphi \mathsf{W} \varphi \mid \exists \varphi \mid \forall \varphi$$

  The quantifiers $\exists$ and $\forall$ range over tefs:

$$(T, \tau) \models \exists \varphi \text{ iff } (T, \tau') \models \varphi \text{ for some tef } \tau' \text{ of } T \text{ s.t. } \tau'(0) = \tau(0),$$
$$(T, \tau) \models \forall \varphi \text{ iff } (T, \tau') \models \varphi \text{ for all tefs } \tau' \text{ of } T \text{ s.t. } \tau'(0) = \tau(0).$$

- $\mathrm{TeamCTL}$ has the same syntax as CTL:

$$\varphi ::= p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathsf{X}_\exists\varphi \mid \mathsf{X}_\forall\varphi \mid \varphi \mathsf{U}_\exists \varphi \mid \varphi \mathsf{U}_\forall \varphi \mid \varphi \mathsf{W}_\exists \varphi \mid \varphi \mathsf{W}_\forall \varphi,$$

  where $\varphi \mathsf{U}_\exists \varphi$ is a shorthand for $\exists \varphi \mathsf{U} \varphi$ etc.

Quantifier extensions of TeamLTL

- $\mathrm{TeamCTL}^*$ has the same syntax as CTL*:

$$\varphi ::= p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \varphi \mathsf{W} \varphi \mid \exists \varphi \mid \forall \varphi$$

  The quantifiers $\exists$ and $\forall$ range over tefs:

$$(T, \tau) \models \exists \varphi \text{ iff } (T, \tau') \models \varphi \text{ for some tef } \tau' \text{ of } T \text{ s.t. } \tau'(0) = \tau(0),$$
$$(T, \tau) \models \forall \varphi \text{ iff } (T, \tau') \models \varphi \text{ for all tefs } \tau' \text{ of } T \text{ s.t. } \tau'(0) = \tau(0).$$

- $\mathrm{TeamCTL}$ has the same syntax as CTL:

$$\varphi ::= p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathsf{X}_\exists \varphi \mid \mathsf{X}_\forall \varphi \mid \varphi \mathsf{U}_\exists \varphi \mid \varphi \mathsf{U}_\forall \varphi \mid \varphi \mathsf{W}_\exists \varphi \mid \varphi \mathsf{W}_\forall \varphi,$$

  where $\varphi \mathsf{U}_\exists \varphi$ is a shorthand for $\exists \varphi \mathsf{U} \varphi$ etc.
- $\exists$TeamCTL* is the fragments of $\mathrm{TeamCTL}$ without the modalities $\{\mathsf{U}_\forall, \mathsf{W}_\forall, \mathsf{X}_\forall\}$
- $\forall\mathrm{TeamCTL}$ is the fragments of $\mathrm{TeamCTL}$ without the modalities $\{\mathsf{U}_\exists, \mathsf{W}_\exists, \mathsf{X}_\exists\}$
- $\exists\mathrm{TeamCTL}^*$, $\forall\mathrm{TeamCTL}^*$ are fragments of $\mathrm{TeamCTL}^*$ without $\forall$ and $\exists$, resp.

Complexity results

Model Checking Problem for	Complexity
$\exists\mathrm{TeamLTL}(\varovee, \subseteq)$	Σ^0_1-hard
$\forall\mathrm{TeamLTL}(\varovee, \subseteq, \mathrm{NE})$	Σ^0_1-hard
$\exists\mathsf{TeamCTL}^*(\varovee, \subseteq)$	Σ^0_1-hard
$\forall\mathrm{TeamCTL}(\varovee, \subseteq)$	Σ^0_1-hard
$\exists\mathsf{TeamCTL}^*(\varovee)$	Σ^1_1-hard
$\mathrm{TeamCTL}^*(\mathcal{S}, \mathrm{ALL})$ for *k*-synchronous or *k*-context-bounded tefs	decidable
$\mathrm{TeamCTL}^*(\mathcal{S})$ for *k*-synchronous or *k*-context-bounded tefs, where *k* and the number of traces is fixed	polynomial time

Table: Complexity results overview. The Σ^0_1-hardness results follow via embeddings of synchronous $\mathrm{TeamLTL}$, whereas the Σ^1_1-hardness truly relies on asynchronity. ALL is the set of all generalised atoms and $\mathcal{S} = \{\varovee, \mathrm{NE}, \overset{1}{\mathrm{A}}, \mathrm{dep}, \subseteq\}$.

Summary

- General framework for temporal team semantics
- We can combine asynchronous and synchronous tefs
- We can embed synchronous TeamLTL
- highly undecidable model-checking problem
- first steps in identifying decidable fragments

Summary

- General framework for temporal team semantics
- We can combine asynchronous and synchronous tefs
- We can embed synchronous TeamLTL
- highly undecidable model-checking problem
- first steps in identifying decidable fragments

Current and future directions

- Indentification of decidable fragments and variants
- Consider tefs also as inputs given in some finite way.

Summary

- General framework for temporal team semantics
- We can combine asynchronous and synchronous tefs
- We can embed synchronous TeamLTL
- highly undecidable model-checking problem
- first steps in identifying decidable fragments

Current and future directions

- Indentification of decidable fragments and variants
- Consider tefs also as inputs given in some finite way.

Thank you!