Linear-time Temporal Logic with Team Semantics: Expressivity and Complexity

Jonni Virtema[1] Jana Hofmann[2]
Bernd Finkbeiner[2] Juha Kontinen[3] Fan Yang[3]

[1] University of Sheffield, UK
[2] CISPA Helmholtz Center for Information Security, Germany
[3] University of Helsinki Finland

15.12.2021 — FSTTCS'21

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.

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.

Snapshot of our paper

State of the art:

- 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.

Snapshot of our paper

State of the art:

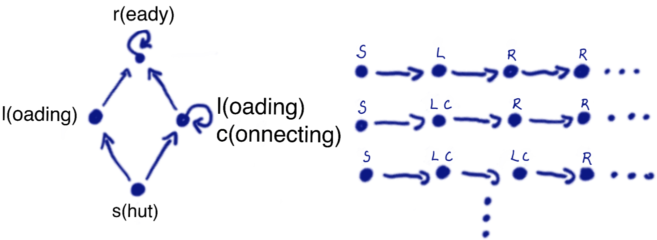
- LTL, QPTL, CTL, etc. vs. HyperLTL, HyperQPTL, HyperCTL, etc.
  are prominent logics for traceproperties vs. hyperproperties of systems
  - Traceproperty: Each request is eventually granted (properties of traces)
  - Hyperproperty: Each request is granted in bounded time (properties of sets of traces)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

This paper:

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

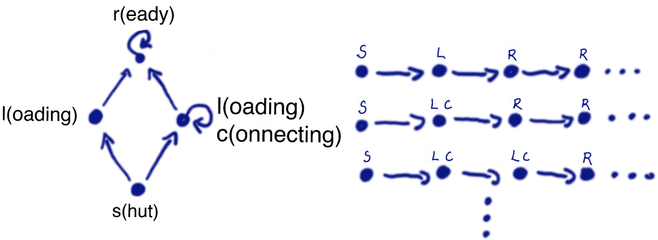
Traceproperties and hyperproperties

Opening your office computer after holidays:



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.

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.
  - LTL is bisimulation invariant (cannot separate systems whose traces behave similarly)
- 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: $\Pi \models_T \varphi$

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

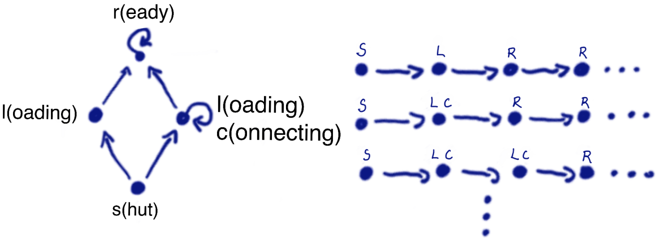
HyperQPTL extends HyperLTL by (uniform) quantification of propositions: $\exists p \varphi$, $\forall p \varphi$

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.

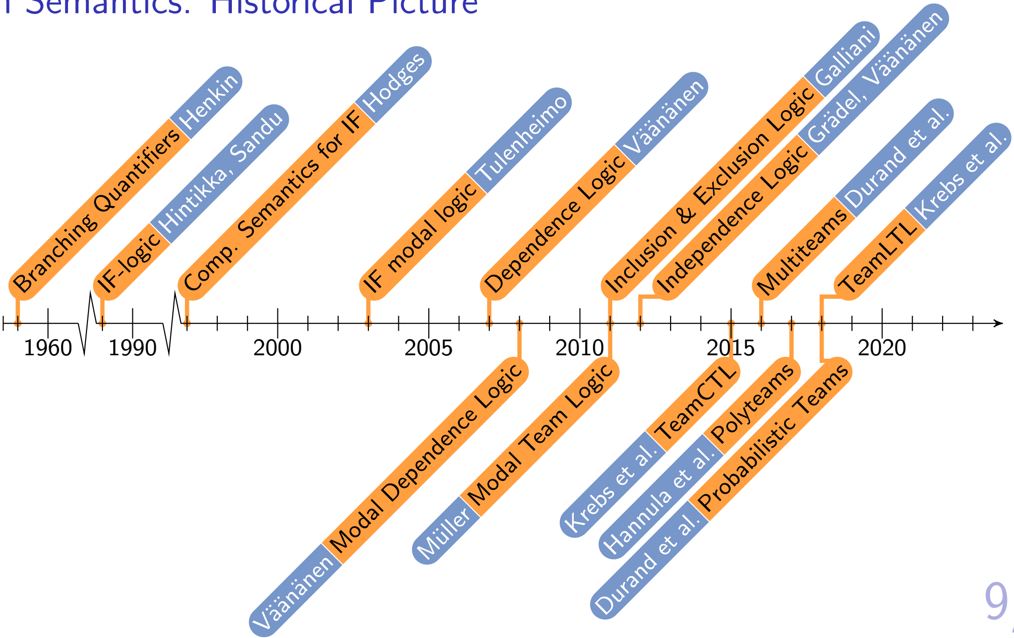
Core of Team Semantics

- 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.

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

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

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

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.

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

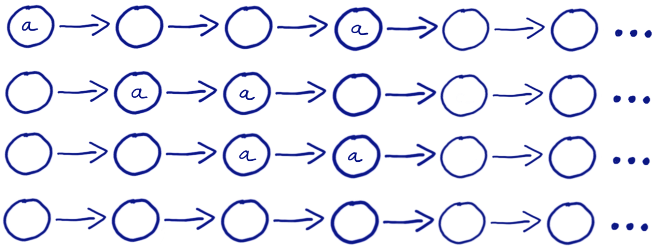
Expressible in synchronous TeamLTL: $\mathsf{FG}\,\neg a$

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: $\mathsf{FG} \neg a$

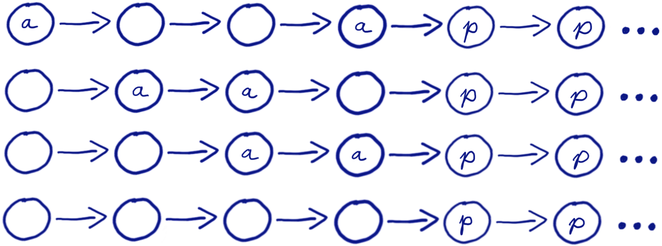


Examples: HyperLTL vs. TeamLTL

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: $\mathsf{FG} \neg a$

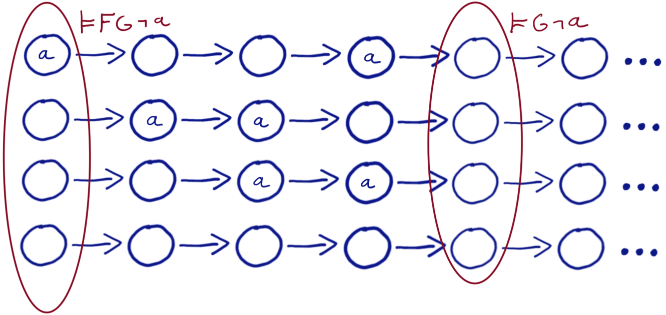


Examples: HyperLTL vs. TeamLTL

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.

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

Examples: HyperLTL vs. TeamLTL

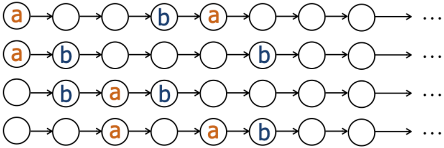
A trace-set T satisfies $\varphi \vee \psi$ if it decomposed to sets T_φ and T_ψ satisfying φ and ψ.

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

$$(T, i) \models \varphi \wedge \psi \text{ iff } (T, i) \models \varphi \text{ and } (T, i) \models \psi$$

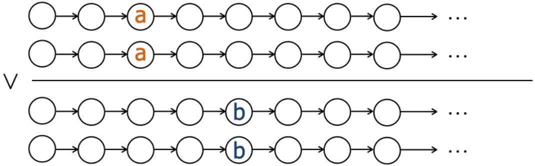
HyperLTL:

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



TeamLTL:

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



Examples: HyperLTL vs. TeamLTL

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

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

Examples: HyperLTL vs. TeamLTL

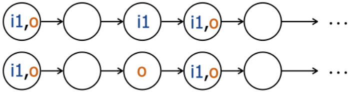
Dependence atom $\mathrm{dep}(p_1, \ldots, p_m, q)$ states that $p_1, \ldots, p_m$ functionally determine q:

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

TeamLTL:

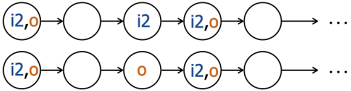
$$(\mathrm{G}\ dep(i1, o)) \vee (\mathrm{G}\ dep(i2, o))$$

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"

Examples: HyperLTL vs. TeamLTL

Boolean disjunction: $(T, i) \models \varphi \varovee \psi$ iff $(T, i) \models \varphi$ or $(T, i) \models \psi$.

Depending on an unknown input, execution traces either agree on a or on b.

Expressible in HyperLTL with three trace quantifiers:

$$\exists \pi_1 \, \exists \pi_2 \, \forall \pi \, \mathsf{G}(a_{\pi_1} \leftrightarrow a_\pi) \vee \mathsf{G}(b_{\pi_2} \leftrightarrow b_\pi).$$

Expressible in TeamLTL:

$$\mathsf{G}\,\mathrm{dep}(a) \vee \mathsf{G}\,\mathrm{dep}(b) \text{ and } \mathsf{G}(a \varovee \neg a) \vee \mathsf{G}(b \varovee \neg b).$$

Quantification of traces in TeamLTL

Inclusion atom $p_1 \dots p_n \subseteq q_1 \dots q_n$ states: truth-values of $p_1 \dots p_n$ occur for $q_1 \dots q_n$.

$$(T, i) \models p_1 \dots p_n \subseteq q_1 \dots q_n \text{ iff } \forall t \exists t' t[i](p_1) = t'[i](q_1), \dots, t[i](p_n) = t'[i](q_n)$$

Inclusion atoms can be used to express traceproperties in TeamLTL:

- $\forall \pi . \varphi_\pi$ can be expressed with $\varphi \subseteq \top$.
- $\exists \pi . \varphi_\pi$ can be expressed with $\top \subseteq \varphi$.

Quantification of traces in TeamLTL

Inclusion atom $p_1 \dots p_n \subseteq q_1 \dots q_n$ states: truth-values of $p_1 \dots p_n$ occur for $q_1 \dots q_n$.

$$(T, i) \models p_1 \dots p_n \subseteq q_1 \dots q_n \text{ iff } \forall t \exists t' t[i](p_1) = t'[i](q_1), \dots, t[i](p_n) = t'[i](q_n)$$

Inclusion atoms can be used to express traceproperties in TeamLTL:

- $\forall \pi . \varphi_\pi$ can be expressed with $\varphi \subseteq \top$.
- $\exists \pi . \varphi_\pi$ can be expressed with $\top \subseteq \varphi$.

Some properties involving single quantifier blocks can be expressed in TeamLTL.

- $\Pi \models_T \forall \pi_1 \dots \forall \pi_n . \varphi_{\vec{\pi}}$ is related to $(T', 0) \models \varphi$ for all subteams $T' \subseteq T$ of size at most n.
- $\Pi \models_T \exists \pi_1 \dots \exists \pi_n . \varphi_{\vec{\pi}}$ is related to $(T', 0) \models \varphi$ for some subteam $T' \subseteq T$ of size at most n.

No obvious way to mimic quantifier alternation without encoding gadgets to traces.

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 $\mathrm{G}\varphi := (\varphi \mathrm{W} \bot)$.

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

Motivation of the current work

- recent interest into temporal team semantics
  [Krebs et al 2018, Lück 2020, Kontinen & Sandsrtöm 2021, Gutsfeld et al. 2021]
- develop purely modal logics for hyperproperties
- discover decidable and expressive logics for hyperproperties
- investigate connections between HyperLTL and TeamLTL variants

Results of our paper

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:

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

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 $S_2 \in B$.
- $(T, i) \models \mathrm{NE}$ 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 [Krebs et al. 2018]
k-coherent $\mathrm{TeamLTL}(\sim)$	in EXPSPACE
left-flat $\mathrm{TeamLTL}(\otimes, \overset{1}{\mathsf{A}})$	in EXPSPACE
$\mathrm{TeamLTL}(\subseteq, \otimes)$	Σ_1^0-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

$$
\begin{array}{lcl}
\mathrm{TeamLTL}(\ovee, \overset{1}{\mathsf{A}}) & \leq & \overset{u}{\exists}{}^{*}_{q}\forall_{\pi}\mathrm{HyperQPTL} \text{ (assuming left-flatness)} \\
 & \leq & \exists_{p}\overset{u}{Q}{}^{*}_{p}\forall_{\pi}\mathrm{HyperQPTL}^{+} \text{ (general case)} \\
\wedge^{\dagger} & & \\
\mathrm{TeamLTL}(\ovee, \mathrm{NE}, \overset{1}{\mathsf{A}}) & \leq & \exists_{p}\overset{u}{Q}{}^{*}_{p}\exists^{*}_{\pi}\forall_{\pi}\mathrm{HyperQPTL}^{+} \\
 & & \\
|\wedge \quad \text{[Luck 2020]} & & \text{(assuming } k\text{-coherence)} \\
\mathrm{TeamLTL}(\sim) & \leq & \forall^{k}\mathrm{HyperLTL}
\end{array}
$$

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

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 $\{j_1, j_2\}$, $\qquad$ C_a^- goto $\{j_1, j_2\}$, $\qquad$ 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 $\{t \in T[i, \infty] \mid t[0](c_a) = 1\}$

Conclusion

- TeamLTL is a promising purely modal alternative for a logic for hyperproperties
- Expressiveness
  - Uncomparable with HyperLTL
  - Assuming left-flatness and downward closure translates to $\overset{u}{\exists}{}^{*}_{q}\forall_{\pi}\mathrm{HyperQPTL}$.
  - In general translates to $\mathrm{HyperQPTL}^{+}$.
- Complexity
  - In EXPSPACE assuming left-flatness and downward closure
  - In EXPSPACE assuming k-coherence
  - TeamLTL($\subseteq, \varovee$) already undecidable
  - TeamLTL($\subseteq, \varovee$, A) highly undecidable

Conclusion

- TeamLTL is a promising purely modal alternative for a logic for hyperproperties
- Expressiveness
  - Uncomparable with HyperLTL
  - Assuming left-flatness and downward closure translates to $\overset{u}{\exists}{}_q^* \forall_\pi \mathrm{HyperQPTL}$.
  - In general translates to $\mathrm{HyperQPTL}^+$.
- Complexity
  - In EXPSPACE assuming left-flatness and downward closure
  - In EXPSPACE assuming k-coherence
  - TeamLTL($\subseteq, \varovee$) already undecidable
  - TeamLTL($\subseteq, \varovee$, A) highly undecidable

Thank you!