Team Semantics for the Specification and Verification of Hyperproperties

Jonni Virtema

Hasselt University, Belgium
jonni.virtema@gmail.com

Joint work with Andreas Krebs[1], Arne Meier[2], and Martin Zimmermann[3]

[1]University of Tübingen, Germany, [2]University of Hanover, Germany, [3]Saarland University, Germany

27th of August, 2018 – MFCS 2018

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.

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: Motivation and History

Logical modelling of uncertainty, imperfect information, and different notions of dependence such as functional dependence and independence, from application fields: statistics (probabilistic independence), database theory (database dependencies), social choice theory (arrows theore), etc.

Historical development:

- Branching quantifiers by Henkin 1959.
  $\begin{pmatrix} \forall x \exists y \\ \forall x' \exists y' \end{pmatrix} \varphi(x, y, x', y')$
- Independence-friendly logic by Hintikka and Sandu 1989.
  $\forall x \exists y \forall x' \exists y' / \{x, y\} \, \varphi(x, y, x', y')$
- Team semantics by Hodges 1997.
- Dependence logic and modal dependence logic by Väänänen 2007.
- Introduction of other dependency notions to team semantics such as inclusion, exclusion, and independence. Galliani, Grädel, Väänänen.
- Team semantics for computational tree logic CTL by Krebs et al.
- Multiteam, polyteam, and probabilistic team semantics by Hannula et al.

Team Semantics: Motivation and History

Logical modelling of uncertainty, imperfect information, and different notions of dependence such as functional dependence and independence, from application fields: statistics (probabilistic independence), database theory (database dependencies), social choice theory (arrows theore), etc.

Historical development:

- Branching quantifiers by Henkin 1959.
- Independence-friendly logic by Hintikka and Sandu 1989.
- Team semantics by Hodges 1997.
- Dependence logic and modal dependence logic by Väänänen 2007.
- Introduction of other dependency notions to team semantics such as inclusion, exclusion, and independence. Galliani, Grädel, Väänänen.
- Team semantics for computational tree logic CTL by Krebs et al.
- Multiteam, polyteam, and probabilistic team semantics by Hannula et al.

Trace Properties and Hyperproperties

- Behaviour of a system can be modelled via execution traces $\vec{t}$.
  - Think of a (infinite) sequence $\vec{t}$, where $t[i]$ is the state of the system at time i.
- Trace properties are sets of traces of the system in question.
  - A system satisfies a trace property if each of its traces has the property.
  - The system terminates eventually is a trace property.
  - The system terminates within a bounded time is not a trace property.
- Hyperproperties by Clarkson and Schneider 2010
  - Hyperproperties are sets of sets of traces.
  - A system satisfies a hyperproperty H if its set of traces belong to H.
  - Every trace property is a hyperproperty.
  - The system terminates within a bounded time is a hyperproperty.
- Hyperproperties are exactly the same as team properties.

Trace Properties and Hyperproperties

- Behaviour of a system can be modelled via execution traces $\vec{t}$.
  - Think of a (infinite) sequence $\vec{t}$, where $t[i]$ is the state of the system at time i.
- Trace properties are sets of traces of the system in question.
  - A system satisfies a trace property if each of its traces has the property.
  - The system terminates eventually is a trace property.
  - The system terminates within a bounded time is not a trace property.
- Hyperproperties by Clarkson and Schneider 2010
  - Hyperproperties are sets of sets of traces.
  - A system satisfies a hyperproperty H if its set of traces belong to H.
  - Every trace property is a hyperproperty.
  - The system terminates within a bounded time is a hyperproperty.
- Hyperproperties are exactly the same as team properties.

LTL and HyperLTL

- Trace properties are typically specified in temporal logics, most prominently in Linear Temporal Logic (LTL).
- Verification of LTL specifications is routinely employed in industrial settings and marks one of the most successful applications of formal methods to real-life problems.
- HyperLTL by Clarkson et al. 2014 is an extension of LTL for specifying hyperproperties.
- In LTL the satisfying object is a trace. Syntax:

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

- In HyperLTL the satisfying object is a set of traces and a trace assignment.

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

LTL and HyperLTL

- Trace properties are typically specified in temporal logics, most prominently in Linear Temporal Logic (LTL).
- Verification of LTL specifications is routinely employed in industrial settings and marks one of the most successful applications of formal methods to real-life problems.
- HyperLTL by Clarkson et al. 2014 is an extension of LTL for specifying hyperproperties.
- In LTL the satisfying object is a trace. Syntax:

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

- In HyperLTL the satisfying object is a set of traces and a trace assignment.

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

Hyperproperties in HyperLTL

- Majority of the information flow properties found in the literature are expressible.
  - Observational determinism: $\forall\pi\forall\pi' \, (\pi[0] =_{\mathrm{in}} \pi'[0]) \rightarrow (\pi[0] =_{\mathrm{out}} \pi'[0])$
  - Noninference (from high security to low security): $\forall\pi\exists\pi' \, (G\lambda_{\pi'}) \wedge \pi =_L \pi'$
  $\lambda =$ "dummy high security information", in/out="input/output", L="low security information"
- Problems about HyperLTL:
  - Bounded termination is not expressible.
  - Satisfiability problem is undecidable.
  - Model checking problem is non-elementary.

Hyperproperties in HyperLTL

- Majority of the information flow properties found in the literature are expressible.
  - Observational determinism: $\forall\pi\forall\pi' \, (\pi[0] =_{\text{in}} \pi'[0]) \rightarrow (\pi[0] =_{\text{out}} \pi'[0])$
  - Noninference (from high security to low security): $\forall\pi\exists\pi' \, (G\lambda_{\pi'}) \wedge \pi =_L \pi'$ $\lambda =$ "dummy high security information", in/out="input/output", L="low security information"
- Problems about HyperLTL:
  - Bounded termination is not expressible.
  - Satisfiability problem is undecidable.
  - Model checking problem is non-elementary.

Team Semantics for Specifying Hyperproperties

- Motivation:
  - High complexity of HyperLTL.
  - Some interesting hyperproperties are not expressible in HyperLTL.
  - Hyperproperties are team properties.
- Starting point:
  - Extensive research on modal team semantics.
  - Team semantics for CTL.

Team Semantics for Specifying Hyperproperties

- Motivation:
  - High complexity of HyperLTL.
  - Some interesting hyperproperties are not expressible in HyperLTL.
  - Hyperproperties are team properties.
- Starting point:
  - Extensive research on modal team semantics.
  - Team semantics for CTL.

Traces and Teams

- A *trace* over a set AP of propositions is an infinite sequence from $\mathcal{P}(AP)^\omega$.
- A *team* is a (potentially infinite) set of traces over some fixed AP.
- Given a trace $t = t(0)t(1)t(2)\cdots$ and $i \geq 0$, we define

$$t[i,\infty) := t(i)t(i+1)t(i+2)\cdots ,$$

which we lift to teams $T \subseteq \mathcal{P}(AP)^\omega$ by defining

$$T[i,\infty) := \{t[i,\infty) \mid t \in T\}.$$

Syntax and Semantics for TeamLTL

Syntax of LTL in negation normal form:

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid X\varphi \mid F\varphi \mid G\varphi \mid \varphi U \varphi \mid \varphi R \varphi.$$

$t \models p$ if $p \in t(0)$,
$t \models \neg p$ if $p \notin t(0)$,
$t \models \psi \wedge \phi$ if $t \models \psi$ and $t \models \phi$,
$t \models \psi \vee \phi$ if $t \models \psi$ or $t \models \phi$,
$t \models \mathrm{X}\varphi$ if $t[1,\infty) \models \varphi$,

$t \models \mathrm{F}\varphi$ if $\exists k \geq 0 : t[k,\infty) \models \varphi$,
$t \models \mathrm{G}\varphi$ if $\forall k \geq 0 : t[k,\infty) \models \varphi$,
$t \models \psi \mathrm{U} \phi$ if $\exists k \geq 0 : t[k,\infty) \models \phi$ and $\forall k' < k : t[k',\infty) \models \psi$.

Syntax and Semantics for TeamLTL

Syntax of teamLTL in negation normal form:

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid X\varphi \mid F\varphi \mid G\varphi \mid \varphi U \varphi \mid \varphi R \varphi.$$

$T \models^{\star} p$ if $\forall t \in T : p \in t(0)$,
$T \models^{\star} \neg p$ if $\forall t \in T : p \notin t(0)$,
$T \models^{\star} \psi \wedge \phi$ if $T \models^{\star} \psi$ and $T \models^{\star} \phi$,
$T \models^{\star} \psi \vee \phi$ if $\exists T_1 \cup T_2 = T$ such that $T_1 \models^{\star} \psi$ and $T_2 \models^{\star} \phi$,
$T \models^{\star} \mathrm{X}\varphi$ if $T[1, \infty) \models^{\star} \varphi$.

Syntax and Semantics for TeamLTL

Syntax of teamLTL in negation normal form:

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid X\varphi \mid F\varphi \mid G\varphi \mid \varphi U \varphi \mid \varphi R \varphi.$$

Synchronous semantics:

$T \models^{s} \mathrm{F}\phi$ if $\exists k \geq 0 : T[k,\infty) \models^{s} \phi$,
$T \models^{s} \mathrm{G}\phi$ if $\forall k \geq 0 : T[k,\infty) \models^{s} \phi$,
$T \models^{s} \psi \mathrm{U} \phi$ if $\exists k \geq 0 : T[k,\infty) \models^{s} \phi$ and $\forall k' < k : T[k',\infty) \models^{s} \psi$.

Syntax and Semantics for TeamLTL

Syntax of teamLTL in negation normal form:

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid X\varphi \mid F\varphi \mid G\varphi \mid \varphi U \varphi \mid \varphi R \varphi.$$

Synchronous semantics:

$T \models^{\mathsf{s}} \mathrm{F}\phi$ if $\exists k \geq 0 : T[k, \infty) \models^{\mathsf{s}} \phi$,

$T \models^{\mathsf{s}} \mathrm{G}\phi$ if $\forall k \geq 0 : T[k, \infty) \models^{\mathsf{s}} \phi$,

$T \models^{\mathsf{s}} \psi \mathrm{U} \phi$ if $\exists k \geq 0 : T[k, \infty) \models^{\mathsf{s}} \phi$ and $\forall k' < k : T[k', \infty) \models^{\mathsf{s}} \psi$.

Asynchronous semantics:

$T \models^{\mathsf{a}} \mathrm{F}\phi$ if $\exists k_t \geq 0$, for each $t \in T$: $\{t[k_t, \infty) \mid t \in T\} \models^{\mathsf{a}} \phi$

$T \models^{\mathsf{a}} \mathrm{G}\phi$ if $\forall k_t \geq 0$, for each $t \in T$: $\{t[k_t, \infty) \mid t \in T\} \models^{\mathsf{a}} \phi$,

$T \models^{\mathsf{a}} \psi \mathrm{U} \phi$ if $\exists k_t \geq 0$, for each $t \in T$: $\{t[k_t, \infty) \mid t \in T\} \models^{\mathsf{a}} \phi$, and
$\forall k'_t < k_t$, for each $t \in T$: $\{t[k'_t, \infty) \mid t \in T\} \models^{\mathsf{a}} \psi$.

Synchronous vs. Asynchronous

Example

Let $T = \{t, t'\}$, where $t = \{p\}\emptyset^\omega$ and $t' = \emptyset\{p\}\emptyset^\omega$. Now

$$T \models^{\mathrm{a}} Fp$$

as we can pick $k_t = 0$ and $k_{t'} = 1$. On the other hand, there is no single k such that $T[k, \infty) \models^{\mathrm{s}} p$ and consequently $T \not\models^{\mathrm{s}} Fp$.

Extensions of TeamLTL

- Asynchronous teamLTL is essentially ordinary LTL:
  $T \models^{a} \varphi \Leftrightarrow \forall t \in T : t \models \varphi$
- Uniform termination is expressible in synchronous teamLTL:
  $Fp_{\mathrm{terminated}}$
- Both semantics are downward closed: $T \models \varphi$ and $T' \subseteq T$ implies $T' \models \varphi$
  - Simple properties are not expressible in teamLTL: $\exists \pi p_{\pi}$
- We consider extensions of teamLTL:
  - Dependence atoms:
    $T \models \mathrm{dep}(\vec{p}, \vec{q})$ iff all $t, s \in T$ that agree on $\vec{p}$ also agree on $\vec{q}$.
  - Contradictory negation: $T \models \sim\varphi$ iff $T \not\models \varphi$.
  - We could consider other atoms: indedendence, incluision, etc.

Extensions of TeamLTL

- Asynchronous teamLTL is essentially ordinary LTL:
  $T \models^{a} \varphi \Leftrightarrow \forall t \in T : t \models \varphi$
- Uniform termination is expressible in synchronous teamLTL:
  $F p_{\mathrm{terminated}}$
- Both semantics are downward closed: $T \models \varphi$ and $T' \subseteq T$ implies $T' \models \varphi$
  - Simple properties are not expressible in teamLTL: $\exists \pi p_{\pi}$
- We consider extensions of teamLTL:
  - Dependence atoms:
    $T \models \mathrm{dep}(\vec{p}, \vec{q})$ iff all $t, s \in T$ that agree on $\vec{p}$ also agree on $\vec{q}$.
  - Contradictory negation: $T \models \sim\varphi$ iff $T \not\models \varphi$.
  - We could consider other atoms: indedendence, incluision, etc.

Synchronous vs. Asynchronous

Example

Let T be a set of traces and $p \in AP$.

$$T \models^{a} G\,\mathrm{dep}(p)$$

expresses that p has constant value in all positions of all traces, i.e., p is globally true or globally false.

$$T \models^{s} G\,\mathrm{dep}(p)$$

expresses that at every time step i (independently) p has a constant value, i.e., at any fixed time step i, p is globally true or globally false.

Expressive Power of Extensions

- TeamLTL(dep) is downward closed.
  - Observational determinism can be expressed: $\mathrm{dep}(\overline{\mathrm{input}}, \overline{\mathrm{output}})$
  - Noninference cannot be expressed.
- TeamLTL($\sim$) is very expressive.
  - In propositional setting, all team properties can be expressed.
  - In modal setting, all first-order definable team-bisimulation closed team properties can be expressed.
  - Subsumes teamLTL(dep).
  - Non-inference can be expressed:
    "All maximal subteams that have a constant value for low security information includes a trace with dummy high security information."
  - Problem: High complexity.

Expressive Power of Extensions

- TeamLTL(dep) is downward closed.
  - Observational determinism can be expressed: $\mathrm{dep}(\overline{\mathrm{input}}, \overline{\mathrm{output}})$
  - Noninference cannot be expressed.
- TeamLTL($\sim$) is very expressive.
  - In propositional setting, all team properties can be expressed.
  - In modal setting, all first-order definable team-bisimulation closed team properties can be expressed.
  - Subsumes teamLTL(dep).
  - Non-inference can be expressed:
    "All maximal subteams that have a constant value for low security information includes a trace with dummy high security information."
  - Problem: High complexity.

Expressive Power of Extensions

- TeamLTL(dep) is downward closed.
  - Observational determinism can be expressed: $\mathrm{dep}(\overline{\mathrm{input}}, \overline{\mathrm{output}})$
  - Noninference cannot be expressed.
- TeamLTL($\sim$) is very expressive.
  - In propositional setting, all team properties can be expressed.
  - In modal setting, all first-order definable team-bisimulation closed team properties can be expressed.
  - Subsumes teamLTL(dep).
  - Non-inference can be expressed:
    "All maximal subteams that have a constant value for low security information includes a trace with dummy high security information."
  - Problem: High complexity.

Decision Problems

Problem: TeamLTL satisfiability.
Input: An LTL formula φ.
Question: Does there exist a non-empty team T such that $T \models \varphi$?

Problem: TeamPathChecking.
Input: An LTL formula φ and a finite set T of ultimately periodic traces.
Question: Does $T \models \varphi$ hold?

Problem: TeamModelChecking.
Input: An LTL formula φ and a finite Kripke structure K.
Question: Does $T(K) \models^{\star} \varphi$ hold?

Complexity Results

	Satisfiability		Path Checking		Model Checking	
	synchronous	asynchronous	synchronous	asynchronous	synchronous	asynchronous
LTL	—— PSPACE [Sistla, Clarke 85] ——		—— in P ——		—— PSPACE [Sistla, Clarke 85] ——	
HyperLTL	undecidable [Finkbeiner, Hahn 2016]		—— in EXPSPACE ——		—— non-elementary [Clarkson et al. 2014] ——	
TeamLTL	PSPACE	PSPACE	PSPACE	in P	PSPACE-hard	PSPACE
TeamLTL(dep)	PSPACE	PSPACE	PSPACE	PSPACE-h	NEXPTIME-h	NEXPTIME-h
TeamLTL($\sim$)	??	??	PSPACE	PSPACE-h	ATIME-ALT(exp, poly)-h	ATIME-ALT(exp, poly)-h

Colour code for teamLTL:

Red results are the main technical results of the paper.

Violet results are corollaries from the red ones.

Blue results are interesting and non-trivial.

Green results follow from known results with minimum effort.

Source of Hardness Proofs

- We obtain PSPACE from reductions from QBF.
- We give reductions from satisfiability and validity of propositional logics with team semantics to model checking of teamLTL, and obtain hardness for NEXPTIME and ATIME-ALT(exp, poly).

Conclusion

- We defined teamLTL as an alternative for hyperLTL.
- The expressive powers of teamLTL and hyperLTL are orthogonal.
- Some interesting hyperproperties can be expressed in synchronous teamLTL, teamLTL(dep), and teamLTL($\sim$).
- TeamLTL has better algorithmic properties than hyperLTL, though this might not hold for teamLTL($\sim$).

Future Work

- Many open question concerning complexity of extensions of teamLTL.
- Study what extensions/fragments of teamLTL can express most interesting hyperproperties, but has still low enough complexity.
  - What atoms should be used?
  - Should we restrict the syntactic form of the formulas?
- Give a natural team semantics to CTL^* and compare it to $\mathrm{HyperCTL}^*$.

Future Work

Thanks!

- Many open question concerning complexity of extensions of teamLTL.
- Study what extensions/fragments of teamLTL can express most interesting hyperproperties, but has still low enough complexity.
  - What atoms should be used?
  - Should we restrict the syntactic form of the formulas?
- Give a natural team semantics to CTL* and compare it to HyperCTL*.