Abstract Team Logic

Martin Lück
Leibniz Universität Hannover

25. Jahrestagung LOGINF, Jena, 22.10.19

Team-based logic

Classical semantics: Evaluate formulas in a single state of a system.

- Propositional assignment
- First-order structure with assignment
- Point in Kripke structure

Team semantics: Evaluate formulas in multiple states of a system.

- Set of propositional assignments
- Set of assignments in a first-order structure
- Set of points in a Kripke structure

These sets are called teams.

Team-based logic

Classical semantics: Evaluate formulas in a single state of a system.

- Propositional assignment
- First-order structure with assignment
- Point in Kripke structure

Team semantics: Evaluate formulas in multiple states of a system.

- Set of propositional assignments
- Set of assignments in a first-order structure
- Set of points in a Kripke structure

These sets are called teams.

Team-based logic

Example team over $\mathbb{N}$ in first-order logic FO:

	x	y	z
s_1	2	1	2
s_2	1	1	1
s_3	1	0	0

Several interpretations:

- Databases: Set of rows in a table
- Epistemic: Set of states consistent with current knowledge
- Stochastic: Probability distribution

Team-based logic

Historically, team semantics was introduced by Hodges (1997) as a compositional semantics for partially ordered quantifiers, e.g.,

$$\forall x \; \exists y/\{x\} \; \varphi.$$

"There exists y, independently of x, such that φ."
$\rightsquigarrow$ makes no sense for single assignment

Team-based logic

Definition (Väänänen (2007))

Dependence logic is the extension of FO by the *dependence atom*

$$=\!(x_1, \ldots, x_n; y).$$

Read: "y depends only on $x_1, \ldots, x_n$"

Note: $=\!(y)$ means that y is constant in the team.

Team-based logic

Definition (Väänänen (2007))

Dependence logic is the extension of FO by the *dependence atom*

$$=\!(x_1, \ldots, x_n; y).$$

Read: "y depends only on $x_1, \ldots, x_n$"

Note: $=\!(y)$ means that y is constant in the team.

Team-based logic

Formally:

$(\mathcal{A}, T) \vDash {=}(\vec{x}; y) \;:\Leftrightarrow\; \forall s, s' \in T:$ if $s(\vec{x}) = s'(\vec{x})$ then $s(y) = s'(y)$.

	x	y	z
s_1	2	4	2
s_2	1	2	1
s_3	1	2	0

Example: ${=}(x; y)$ is true in this team, but ${=}(y; z)$ is false.

Team-based logic

Formally:

$(\mathcal{A}, T) \models =(\vec{x}; y)$ $:\Leftrightarrow$ $\forall s, s' \in T:$ if $s(\vec{x}) = s'(\vec{x})$ then $s(y) = s'(y)$.

	x	y	z
s_1	2	4	2
s_2	1	2	1
s_3	1	2	0

Example: $=(x; y)$ is true in this team, but $=(y; z)$ is false.

Dependence logic

Dependence logic = FO + dependence atom.
Formulas are in negation normal form.

$$\varphi ::= \ell \mid {=}(\vec{x}; y) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \forall x \varphi$$

Here: ℓ first-order literal.

Let $\mathcal{A}$ be a first-order structure and $T \in \wp(\text{Var} \to \mathcal{A})$.

$(\mathcal{A}, T) \vDash \ell$ $:\Leftrightarrow \forall s \in T : (\mathcal{A}, s) \vDash \ell$, ℓ first-order literal,
$(\mathcal{A}, T) \vDash \varphi \wedge \psi :\Leftrightarrow (\mathcal{A}, T) \vDash \varphi$ and $(\mathcal{A}, T) \vDash \psi$,
$(\mathcal{A}, T) \vDash \varphi \vee \psi :\Leftrightarrow \exists S, U \subseteq T$ such that $T = S \cup U$,
$(\mathcal{A}, S) \vDash \varphi$, and $(\mathcal{A}, U) \vDash \psi$,
(to be continued)

Dependence logic

Dependence logic = FO + dependence atom.
Formulas are in negation normal form.

$$\varphi ::= \ell \mid =(\vec{x}; y) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \forall x \varphi$$

Here: ℓ first-order literal.

Let $\mathcal{A}$ be a first-order structure and $T \in \wp(\mathsf{Var} \to \mathcal{A})$.

$(\mathcal{A}, T) \models \ell \quad :\Leftrightarrow \forall s \in T : (\mathcal{A}, s) \models \ell$, ℓ first-order literal,

$(\mathcal{A}, T) \models \varphi \wedge \psi :\Leftrightarrow (\mathcal{A}, T) \models \varphi$ and $(\mathcal{A}, T) \models \psi$,

$(\mathcal{A}, T) \models \varphi \vee \psi :\Leftrightarrow \exists S, U \subseteq T$ such that $T = S \cup U$,

$(\mathcal{A}, S) \models \varphi$, and $(\mathcal{A}, U) \models \psi$,

(to be continued)

Dependence logic

Dependence logic = FO + dependence atom.
Formulas are in negation normal form.

$$\varphi ::= \ell \mid {=}(\vec{x}; y) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \forall x \varphi$$

Here: ℓ first-order literal.

Let $\mathcal{A}$ be a first-order structure and $T \in \wp(\mathsf{Var} \to \mathcal{A})$.

$(\mathcal{A}, T) \models \ell \quad :\Leftrightarrow \forall s \in T : (\mathcal{A}, s) \models \ell$, ℓ first-order literal,
$(\mathcal{A}, T) \models \varphi \wedge \psi :\Leftrightarrow (\mathcal{A}, T) \models \varphi$ and $(\mathcal{A}, T) \models \psi$,
$(\mathcal{A}, T) \models \varphi \vee \psi :\Leftrightarrow \exists S, U \subseteq T$ such that $T = S \cup U$,
$(\mathcal{A}, S) \models \varphi$, and $(\mathcal{A}, U) \models \psi$,

(to be continued)

Dependence logic

Dependence logic = FO + dependence atom.
Formulas are in negation normal form.

$$\varphi ::= \ell \mid =(\vec{x}; y) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \forall x \varphi$$

Here: ℓ first-order literal.

Let $\mathcal{A}$ be a first-order structure and $T \in \wp(\mathsf{Var} \to \mathcal{A})$.

$(\mathcal{A}, T) \vDash \ell \quad :\Leftrightarrow \forall s \in T : (\mathcal{A}, s) \vDash \ell$, ℓ first-order literal,
$(\mathcal{A}, T) \vDash \varphi \wedge \psi :\Leftrightarrow (\mathcal{A}, T) \vDash \varphi$ and $(\mathcal{A}, T) \vDash \psi$,
$(\mathcal{A}, T) \vDash \varphi \vee \psi :\Leftrightarrow \exists S, U \subseteq T$ such that $T = S \cup U$,
$(\mathcal{A}, S) \vDash \varphi$, and $(\mathcal{A}, U) \vDash \psi$,
(to be continued)

Dependence logic

Dependence logic = FO + dependence atom.
Formulas are in negation normal form.

$$\varphi ::= \ell \mid =\!(\vec{x}; y) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \forall x \varphi$$

Here: ℓ first-order literal.

Let $\mathcal{A}$ be a first-order structure and $T \in \wp(\mathsf{Var} \to \mathcal{A})$.

$(\mathcal{A}, T) \models \ell \quad :\Leftrightarrow \forall s \in T : (\mathcal{A}, s) \models \ell$, ℓ first-order literal,
$(\mathcal{A}, T) \models \varphi \wedge \psi :\Leftrightarrow (\mathcal{A}, T) \models \varphi$ and $(\mathcal{A}, T) \models \psi$,
$(\mathcal{A}, T) \models \varphi \vee \psi :\Leftrightarrow \exists S, U \subseteq T$ such that $T = S \cup U$,
$(\mathcal{A}, S) \models \varphi$, and $(\mathcal{A}, U) \models \psi$,

(to be continued)

Semantics of $\vee$

Disjunction in team semantics:

$$(\mathcal{A}, T) \vDash \varphi \vee \psi :\Leftrightarrow \exists S, U \subseteq T \text{ such that } T = S \cup U, \\ (\mathcal{A}, S) \vDash \varphi, \text{ and } (\mathcal{A}, U) \vDash \psi,$$

	x	y	z	w
s_1	0	1	1	0
s_2	2	2	1	4
s_3	4	2	0	5

- $(\mathbb{N}, T) \vDash (x < 1) \vee (w > 3)$
- $(\mathbb{N}, T) \nvDash =\!(x) \vee =\!(x)$

Semantics of $\vee$

Disjunction in team semantics:

$$(\mathcal{A}, T) \models \varphi \vee \psi :\Leftrightarrow \exists S, U \subseteq T \text{ such that } T = S \cup U, \\ (\mathcal{A}, S) \models \varphi, \text{ and } (\mathcal{A}, U) \models \psi,$$

	x	y	z	w	
s_1	0	1	1	0	S
s_2	2	2	1	4	U
s_3	4	2	0	5	

- $(\mathbb{N}, T) \models (x < 1) \vee (w > 3)$
- $(\mathbb{N}, T) \not\models {=}(x) \vee {=}(x)$

Semantics of $\vee$

Disjunction in team semantics:

$$(\mathcal{A}, T) \vDash \varphi \vee \psi :\Leftrightarrow \exists S, U \subseteq T \text{ such that } T = S \cup U, \\ (\mathcal{A}, S) \vDash \varphi, \text{ and } (\mathcal{A}, U) \vDash \psi,$$

	x	y	z	w
s_1	0	1	1	0
s_2	2	2	1	4
s_3	4	2	0	5

- $(\mathbb{N}, T) \vDash (x < 1) \vee (w > 3)$
- $(\mathbb{N}, T) \not\vDash {=}(x) \vee {=}(x)$

Semantics of ∃

We call $f : T \to \wp^+(\mathcal{A})$ *supplementing function.*

Example: $f(s_1) = \{5\}$ and $f(s_2) = \{5, 6, 7\}$.

	y
s_1	0
s_2	1

$\Rightarrow$

	y	x
$s_{1,1}$	0	5
$s_{2,1}$	1	5
$s_{2,2}$	1	6
$s_{2,3}$	1	7

- $(\mathcal{A}, T) \vDash \exists x\, \varphi \;\Leftrightarrow\; \exists f : T \to \wp^+(\mathcal{A})$ such that $(\mathcal{A}, T^x_f) \vDash \varphi$

Semantics of $\exists$

We call $f : T \to \wp^+(\mathcal{A})$ *supplementing function*.

Example: $f(s_1) = \{5\}$ and $f(s_2) = \{5, 6, 7\}$.

	y
s_1	0
s_2	1

$\Rightarrow$

	y	x
$s_{1,1}$	0	5
$s_{2,1}$	1	5
$s_{2,2}$	1	6
$s_{2,3}$	1	7

- $(\mathcal{A}, T) \vDash \exists x\, \varphi \;\Leftrightarrow\; \exists f : T \to \wp^+(\mathcal{A})$ such that $(\mathcal{A}, T_f^x) \vDash \varphi$

Semantics of $\exists$

We call $f : T \to \wp^+(\mathcal{A})$ *supplementing function*.

Example: $f(s_1) = \{5\}$ and $f(s_2) = \{5, 6, 7\}$.

	y
s_1	0
s_2	1

$\Rightarrow$

	y	x
$s_{1,1}$	0	5
$s_{2,1}$	1	5
$s_{2,2}$	1	6
$s_{2,3}$	1	7

- $(\mathcal{A}, T) \models \exists x\, \varphi \;\Leftrightarrow\; \exists f : T \to \wp^+(\mathcal{A})$ such that $(\mathcal{A}, T^x_f) \models \varphi$

Semantics of ∀

	y
s_1	0
s_2	1

$\Rightarrow$

	y	x
$s_{1,0}$	0	0
$s_{1,1}$	0	1
$s_{1,2}$	0	2
$\vdots$		
$s_{2,0}$	1	0
$s_{2,1}$	1	1
$s_{2,2}$	1	2
$\vdots$		

- $(\mathcal{A}, T) \vDash \forall x\, \varphi \;\Leftrightarrow\; (\mathcal{A}, T^x_{\mathcal{A}}) \vDash \varphi$

Boolean negation

Sometimes the Boolean negation $(\sim)$ is added:

$$(\mathcal{A}, T) \vDash \sim\varphi \quad :\Leftrightarrow \quad (\mathcal{A}, T) \nvDash \varphi$$

$\neg$ is not the Boolean negation in team semantics! It is possible that, for example, $T \nvDash x = 1$ and $T \nvDash \neg(x = 1)$.

Flatness

Classical formulas obey flatness:

$$(\mathcal{A}, T) \models \varphi \quad \Leftrightarrow \quad \forall s \in T : (\mathcal{A}, \{s\}) \models \varphi$$

for $\varphi \in$ FO.
(Proof: By induction.)

For example $=\!(x)$ is not flat.

Flatness

Classical formulas obey flatness:

$$(\mathcal{A}, T) \vDash \varphi \quad \Leftrightarrow \quad \forall s \in T : (\mathcal{A}, \{s\}) \vDash \varphi$$

for $\varphi \in \mathsf{FO}$.
(Proof: By induction.)

For example $=\!(x)$ is not flat.

Motivation

Based on team semantics, numerous logics have been defined:

- Propositional ...		... dependence $(=(x;y))$ ...		
- Modal ...	$+$	... independence $(x \perp y)$ ...	$+$	... logic.
- First-order ...		... inclusion $(x \subseteq y)$ ...		

Many papers on expressiveness and complexity.

Logic	Satisfiability	Validity
$\mathsf{PL}(=(\cdots))$	NP	NEXP
$\mathsf{ML}(=(\cdots))$	NEXP	NEXP
$\mathsf{PL}(\perp)$	NP	NEXP-hard, in Π_2^E
$\mathsf{ML}(\perp)$	NEXP	Π_2^E-hard
$\mathsf{PL}(\subseteq)$	EXP	co-NP
$\mathsf{ML}(\subseteq)$	EXP	co-NEXP-hard

Motivation

Based on team semantics, numerous logics have been defined:

■ Propositional …		… dependence $(=(x;y))$ …		
■ Modal …	$+$	… independence $(x \perp y)$ …	$+$	… logic.
■ First-order …		… inclusion $(x \subseteq y)$ …		

Many papers on expressiveness and complexity.

This talk: Abstract framework to study team logic in.

- Fundamental principles for definition of team semantics?
- "Teamify" other connectives or even logics?
- Formulate results instead of re-proving from scratch everytime?
  - locality, various closure properties, expressiveness results …

Motivation

Based on team semantics, numerous logics have been defined:

▪ Propositional ...		... dependence ($=(x;y)$) ...		
▪ Modal ...	$+$	... independence ($x \perp y$) ...	$+$	... logic.
▪ First-order ...		... inclusion ($x \subseteq y$) ...		

Many papers on expressiveness and complexity.

This talk: Abstract framework to study team logic in.

- Fundamental principles for definition of team semantics?
- "Teamify" other connectives or even logics?
- Formulate results instead of re-proving from scratch everytime?
  - locality, various closure properties, expressiveness results ...

Motivation

Based on team semantics, numerous logics have been defined:

- Propositional ...
- Modal ...
- First-order ...

$+$

... dependence ($=(x;y)$) ...
... independence ($x \perp y$) ...
... inclusion ($x \subseteq y$) ...

$+$ ... logic.

Many papers on expressiveness and complexity.

This talk: Abstract framework to study team logic in.

- Fundamental principles for definition of team semantics?
- "Teamify" other connectives or even logics?
- Formulate results instead of re-proving from scratch everytime?
  - locality, various closure properties, expressiveness results ...

Motivation

Based on team semantics, numerous logics have been defined:

■ Propositional …		… dependence ($=(x;y)$) …		
■ Modal …	$+$	… independence ($x \perp y$) …	$+$	… logic.
■ First-order …		… inclusion ($x \subseteq y$) …		

Many papers on expressiveness and complexity.

This talk: Abstract framework to study team logic in.

- Fundamental principles for definition of team semantics?
- "Teamify" other connectives or even logics?
- Formulate results instead of re-proving from scratch everytime?
  - locality, various closure properties, expressiveness results …

Algebraic definition of semantics

Abstract definition in terms of universal algebra.

Definition

A signature τ is a set of symbols, e.g., $\tau = \{\neg, \wedge, \vee\} \cup \mathsf{Prop}$.

Definition

A *τ-algebra* $\mathfrak{A} = (A, (f_\triangle)_{\triangle \in \tau})$ consists of

- a non-empty set A, the *carrier*, and
- for each $\triangle \in \tau$ a map $f_\triangle \colon A^{\mathrm{arity}(\triangle)} \to A$.

Think of elements $a \in A$ as properties.

Usually, we want $A = \wp U$ for some "universe" U, such that, e.g., $f_\wedge = \cap$ and $f_\vee = \cup$.

Algebraic definition of semantics

Abstract definition in terms of universal algebra.

Definition

A signature τ is a set of symbols, e.g., $\tau = \{\neg, \wedge, \vee\} \cup \mathsf{Prop}$.

Definition

A *τ-algebra* $\mathfrak{A} = (A, (f_\triangle)_{\triangle \in \tau})$ consists of

- a non-empty set A, the *carrier*, and
- for each $\triangle \in \tau$ a map $f_\triangle \colon A^{\mathrm{arity}(\triangle)} \to A$.

Think of elements $a \in A$ as properties.

Usually, we want $A = \wp U$ for some "universe" U, such that, e.g., $f_\wedge = \cap$ and $f_\vee = \cup$.

Algebraic definition of semantics

Abstract definition in terms of universal algebra.

Definition

A signature τ is a set of symbols, e.g., $\tau = \{\neg, \wedge, \vee\} \cup \mathsf{Prop}$.

Definition

A *τ-algebra* $\mathfrak{A} = (A, (f_\triangle)_{\triangle \in \tau})$ consists of

- a non-empty set A, the *carrier*, and
- for each $\triangle \in \tau$ a map $f_\triangle \colon A^{\mathrm{arity}(\triangle)} \to A$.

Think of elements $a \in A$ as properties.

Usually, we want $A = \wp U$ for some "universe" U, such that, e.g., $f_\wedge = \cap$ and $f_\vee = \cup$.

Algebraic definition of semantics

Definition

A *τ-team algebra* is a τ-algebra $\mathfrak{T} = (A, (f_\triangle)_{\triangle\in\tau})$ where $A = \wp\wp U$ for some set U.

Properties a are *sets of teams*, that is, sets of sets.

Obtain "natural" $g\colon (\wp\wp U)^n \to \wp\wp U$ from $f\colon (\wp U)^n \to \wp U$?

Algebraic definition of semantics

Definition

A *τ-team algebra* is a τ-algebra $\mathfrak{T} = (A, (f_\triangle)_{\triangle\in\tau})$ where $A = \wp\wp U$ for some set U.

Properties a are *sets of teams*, that is, sets of sets.

Obtain "natural" $g\colon (\wp\wp U)^n \to \wp\wp U$ from $f\colon (\wp U)^n \to \wp U$?

Teamification

Let $f\colon (\wp U)^n \to \wp U$ and $g\colon (\wp\wp U)^n \to \wp\wp U$.

Definition

We call g a teamification of f if

$$\wp f(T_1, \ldots, T_n) = g(\wp T_1, \ldots, \wp T_n)$$

for all $T_1, \ldots, T_n \subseteq U$.

If this holds for all connectives, then $\wp$ is an algebra homomorphism.

Teamification

Theorem

A map $g\colon (\wp\wp U)^n \to \wp\wp U$ is a teamification iff it preserves flatness.

(A property $a \in \wp\wp U$ is flat if $T \in a \Leftrightarrow \forall s \in T : \{s\} \in a$.)

(Flatness preserving: $a_1, \ldots, a_n$ flat $\Rightarrow g(a_1, \ldots, a_n)$ flat.)

Theorem

Team-semantical $\wedge, \vee, \exists x, \forall x$ are teamifications of classical counterparts.

Same for $\wedge, \vee, \Diamond, \Box$ in modal team logic, and $\wedge, \vee$ in propositional team logic.

Teamification

Theorem

A map $g\colon (\wp\wp U)^n \to \wp\wp U$ is a teamification iff it preserves flatness.

(A property $a \in \wp\wp U$ is flat if $T \in a \Leftrightarrow \forall s \in T : \{s\} \in a$.)

(Flatness preserving: $a_1, \ldots, a_n$ flat $\Rightarrow g(a_1, \ldots, a_n)$ flat.)

Theorem

Team-semantical $\wedge, \vee, \exists x, \forall x$ are teamifications of classical counterparts.

Same for $\wedge, \vee, \Diamond, \Box$ in modal team logic, and $\wedge, \vee$ in propositional team logic.

Teamification

Theorem

A map $g\colon (\wp\wp U)^n \to \wp\wp U$ is a teamification iff it preserves flatness.

(A property $a \in \wp\wp U$ is flat if $T \in a \Leftrightarrow \forall s \in T : \{s\} \in a$.)

(Flatness preserving: $a_1, \ldots, a_n$ flat $\Rightarrow g(a_1, \ldots, a_n)$ flat.)

Theorem

Team-semantical $\wedge, \vee, \exists x, \forall x$ are teamifications of classical counterparts.

Same for $\wedge, \vee, \Diamond, \Box$ in modal team logic, and $\wedge, \vee$ in propositional team logic.

Teamification

Example: Getting rid of negation normal form.

How to define $\neg$ on teams?

$$T \models \neg\varphi \quad :\Leftrightarrow \quad S \not\models \varphi \text{ for all non-empty } S \subseteq T$$

is a teamification of classical negation $\neg$.

Proof.

$$\begin{aligned}
\wp f_{\neg}(T) &= \wp(U \setminus T) \\
&= \{T' \subseteq U \mid T' \cap T = \emptyset\} \\
&= \{T' \subseteq U \mid \forall S \subseteq T' : S \cap T = \emptyset\} \\
&= \{T' \subseteq U \mid \forall S \subseteq T' : \wp S \cap \wp T = \{\emptyset\}\} \\
&= \{T' \subseteq U \mid \forall S \subseteq T' : S = \emptyset \text{ or } S \notin \wp T\} \\
&= g_{\neg}(\wp T)
\end{aligned}$$

□

Teamification

Example: Getting rid of negation normal form.

How to define $\neg$ on teams?

$$T \vDash \neg\varphi \;:\Leftrightarrow\; S \nvDash \varphi \text{ for all non-empty } S \subseteq T$$

is a teamification of classical negation $\neg$.

Proof.

$$\begin{aligned}
\wp f_{\neg}(T) &= \wp(U \setminus T) \\
&= \{T' \subseteq U \mid T' \cap T = \emptyset\} \\
&= \{T' \subseteq U \mid \forall S \subseteq T' : S \cap T = \emptyset\} \\
&= \{T' \subseteq U \mid \forall S \subseteq T' : \wp S \cap \wp T = \{\emptyset\}\} \\
&= \{T' \subseteq U \mid \forall S \subseteq T' : S = \emptyset \text{ or } S \notin \wp T\} \\
&= g_{\neg}(\wp T)
\end{aligned}$$

□

Teamification

Problem: Teamifications are fully determined on flat arguments, but they are arbitrary on non-flat arguments!

Formulas such as $=(x)$ are non-flat.

Theorem

Let g_i be a teamification of f_i, $i \in \{1,2\}$. Then the following statements are equivalent:

- $f_1 = f_2$.
- $g_1(\vec{a}) = g_2(\vec{a})$ *for all flat arguments* $\vec{a}$.

Teamification

Problem: Teamifications are fully determined on flat arguments, but they are arbitrary on non-flat arguments!

Formulas such as $=(x)$ are non-flat.

Theorem

Let g_i be a teamification of f_i, $i \in \{1, 2\}$. Then the following statements are equivalent:

- $f_1 = f_2$.
- $g_1(\vec{a}) = g_2(\vec{a})$ *for all flat arguments* $\vec{a}$.

Operators

Further sensible restrictions for connectives on arbitrary arguments?

- The connectives $\wedge, \vee, \exists x, \forall x, \Diamond, \Box$ are operators in the sense of Jónsson and Tarski (1951).
- Every n-ary operator $\triangle$ is induced by some $(n+1)$-ary "successor" relation $R_\triangle$:
  $T \vDash \triangle(\varphi_1, \ldots, \varphi_n) :\Leftrightarrow \exists (T, S_1, \ldots, S_n) \in R_\triangle$ and $S_i \vDash \varphi_i$ for all i
- Unlike in classical logic, $\wedge$, $\forall x$ and $\Box$ are operators in team semantics!
- Also, $\sim\neg$ is an operator.

Natural class $\mathcal{C}$ $\subseteq$ Operators $\cap$ Teamifications ?

Operators

Further sensible restrictions for connectives on arbitrary arguments?

- The connectives $\wedge, \vee, \exists x, \forall x, \Diamond, \Box$ are operators in the sense of Jónsson and Tarski (1951).
- Every n-ary operator $\triangle$ is induced by some $(n+1)$-ary "successor" relation $R_\triangle$:
  $T \vDash \triangle(\varphi_1, \ldots, \varphi_n) :\Leftrightarrow \exists(T, S_1, \ldots, S_n) \in R_\triangle$ and $S_i \vDash \varphi_i$ for all i
- Unlike in classical logic, $\wedge$, $\forall x$ and $\Box$ are operators in team semantics!
- Also, $\sim\neg$ is an operator.

Natural class $\mathcal{C}$ $\subseteq$ Operators $\cap$ Teamifications ?

Operators

Further sensible restrictions for connectives on arbitrary arguments?

- The connectives $\wedge, \vee, \exists x, \forall x, \Diamond, \Box$ are operators in the sense of Jónsson and Tarski (1951).
- Every n-ary operator $\triangle$ is induced by some $(n+1)$-ary "successor" relation $R_\triangle$:
  $T \vDash \triangle(\varphi_1, \ldots, \varphi_n) :\Leftrightarrow \exists(T, S_1, \ldots, S_n) \in R_\triangle$ and $S_i \vDash \varphi_i$ for all i
- Unlike in classical logic, $\wedge$, $\forall x$ and $\Box$ are operators in team semantics!
- Also, $\sim\neg$ is an operator.

Natural class $\mathcal{C}$ $\subseteq$ Operators $\cap$ Teamifications ?

Operators

Further sensible restrictions for connectives on arbitrary arguments?

- The connectives $\wedge, \vee, \exists x, \forall x, \Diamond, \Box$ are operators in the sense of Jónsson and Tarski (1951).
- Every n-ary operator $\triangle$ is induced by some $(n+1)$-ary "successor" relation $R_\triangle$:
  $T \vDash \triangle(\varphi_1, \ldots, \varphi_n) :\Leftrightarrow \exists(T, S_1, \ldots, S_n) \in R_\triangle$ and $S_i \vDash \varphi_i$ for all i
- Unlike in classical logic, $\wedge$, $\forall x$ and $\Box$ are operators in team semantics!
- Also, $\sim\neg$ is an operator.

Natural class $\mathcal{C}$ $\subseteq$ Operators $\cap$ Teamifications ?

Operators

Further sensible restrictions for connectives on arbitrary arguments?

- The connectives $\wedge, \vee, \exists x, \forall x, \Diamond, \Box$ are operators in the sense of Jónsson and Tarski (1951).
- Every n-ary operator $\triangle$ is induced by some $(n+1)$-ary "successor" relation $R_\triangle$:
  $T \vDash \triangle(\varphi_1, \ldots, \varphi_n) :\Leftrightarrow \exists (T, S_1, \ldots, S_n) \in R_\triangle$ and $S_i \vDash \varphi_i$ for all i
- Unlike in classical logic, $\wedge$, $\forall x$ and $\Box$ are operators in team semantics!
- Also, $\sim\neg$ is an operator.

Natural class $\mathcal{C}$ $\subseteq$ Operators $\cap$ Teamifications ?

Transversals

Definition

A unary operator $\triangle$ is a *transversal* if, for all teams T, S,

$$R_\triangle(T,S) \Leftrightarrow S = \bigcup_{w \in T} f(w) \text{ for some } f \text{ such that } R_\triangle(\{w\}, f(w))$$

Successors of T are precisely the unions of successors of singletons in T. (Definition for n-ary transversals similar.)

Theorem

Transversals are flatness preserving.

Theorem

The connectives $\wedge, \vee, \exists x, \forall x, \Diamond, \Box$ are transversals, as well as all "classical" atomic formulas in team semantics.

Transversals

Definition

A unary operator $\triangle$ is a *transversal* if, for all teams T, S,

$$R_\triangle(T, S) \iff S = \bigcup_{w \in T} f(w) \text{ for some } f \text{ such that } R_\triangle(\{w\}, f(w))$$

Successors of T are precisely the unions of successors of singletons in T.
(Definition for n-ary transversals similar.)

Theorem

Transversals are flatness preserving.

Theorem

The connectives $\wedge, \vee, \exists x, \forall x, \Diamond, \Box$ are transversals, as well as all "classical" atomic formulas in team semantics.

Transversals

Definition

A unary operator $\triangle$ is a *transversal* if, for all teams T, S,

$$R_\triangle(T, S) \;\Leftrightarrow\; S = \bigcup_{w \in T} f(w) \text{ for some } f \text{ such that } R_\triangle(\{w\}, f(w))$$

Successors of T are precisely the unions of successors of singletons in T.
(Definition for n-ary transversals similar.)

Theorem

Transversals are flatness preserving.

Theorem

The connectives $\wedge, \vee, \exists x, \forall x, \Diamond, \Box$ are transversals, as well as all "classical" atomic formulas in team semantics.

Transversals

Definition

A unary operator $\triangle$ is a *transversal* if, for all teams T, S,

$$R_\triangle(T,S) \iff S = \bigcup_{w \in T} f(w) \text{ for some } f \text{ such that } R_\triangle(\{w\}, f(w))$$

Successors of T are precisely the unions of successors of singletons in T.
(Definition for n-ary transversals similar.)

Theorem

Transversals are flatness preserving.

Theorem

The connectives $\wedge, \vee, \exists x, \forall x, \Diamond, \Box$ are transversals, as well as all "classical" atomic formulas in team semantics.

Transversals

Definition

A unary operator $\triangle$ is a *transversal* if, for all teams T, S,

$$R_\triangle(T, S) \;\Leftrightarrow\; S = \bigcup_{w \in T} f(w) \text{ for some } f \text{ such that } R_\triangle(\{w\}, f(w))$$

Successors of T are precisely the unions of successors of singletons in T.
(Definition for n-ary transversals similar.)

Theorem

Transversals are flatness preserving.

Theorem

The connectives $\wedge, \vee, \exists x, \forall x, \Diamond, \Box$ are transversals, as well as all "classical" atomic formulas in team semantics.

Transversals

- Conjunction $\wedge$: $R_{\wedge}\{s\} = \big\{\{s\}\{s\}\big\}$
- Disjunction $\vee$: $R_{\vee}\{s\} = \big\{\{s\}\{s\}, \emptyset\{s\}, \{s\}\emptyset\big\}$
- Existential quantifier $\exists x$: $R_{\exists x}\{s\} = \wp^{+}\{s_a^x \mid a \in \mathcal{A}\}$
- Universal quantifier $\forall x$: $R_{\forall x}\{s\} = \big\{\{s_a^x \mid a \in \mathcal{A}\}\big\}$

Transversals

- Conjunction $\wedge$: $R_{\wedge}\{s\} = \big\{\{s\}\{s\}\big\}$
- Disjunction $\vee$: $R_{\vee}\{s\} = \big\{\{s\}\{s\}, \emptyset\{s\}, \{s\}\emptyset\big\}$
- Existential quantifier $\exists x$: $R_{\exists x}\{s\} = \wp^{+}\{s_a^x \mid a \in \mathcal{A}\}$
- Universal quantifier $\forall x$: $R_{\forall x}\{s\} = \big\{\{s_a^x \mid a \in \mathcal{A}\}\big\}$

Transversals

- Conjunction $\wedge$: $R_\wedge\{s\} = \big\{\{s\}\{s\}\big\}$
- Disjunction $\vee$: $R_\vee\{s\} = \big\{\{s\}\{s\}, \emptyset\{s\}, \{s\}\emptyset\big\}$
- Existential quantifier $\exists x$: $R_{\exists x}\{s\} = \wp^+\{s_a^x \mid a \in \mathcal{A}\}$
- Universal quantifier $\forall x$: $R_{\forall x}\{s\} = \big\{\{s_a^x \mid a \in \mathcal{A}\}\big\}$

Transversals

- Conjunction $\wedge$: $R_{\wedge}\{s\} = \big\{\{s\}\{s\}\big\}$
- Disjunction $\vee$: $R_{\vee}\{s\} = \big\{\{s\}\{s\}, \emptyset\{s\}, \{s\}\emptyset\big\}$
- Existential quantifier $\exists x$: $R_{\exists x}\{s\} = \wp^{+}\{s_a^x \mid a \in \mathcal{A}\}$
- Universal quantifier $\forall x$: $R_{\forall x}\{s\} = \big\{\{s_a^x \mid a \in \mathcal{A}\}\big\}$

Transversals

Application:

Theorem

Every $\mathsf{FO}(\sim)$*-formula is equivalent to a Boolean combination of* FO*-formulas.*

Proof idea: Transversals* commute** with all Boolean operators!

Theorem

Satisfiability of the team-semantical extensions

- $\mathsf{FO}^2(\sim)$ *of two-variable first-order logic* FO^2,
- $\mathsf{GF}(\sim)$ *of the guarded fragment* GF *of first-order logic,*
- $\mathsf{ML}(\sim)$ *of modal logic* ML

is decidable.

In fact logspace-complete for the class $\mathrm{TIME}\left(\exp^{\mathrm{poly}(n)}(1)\right)$.

Transversals

Application:

Theorem

Every FO($\sim$)*-formula is equivalent to a Boolean combination of* FO*-formulas.*

Proof idea: Transversals* commute** with all Boolean operators!

Theorem

Satisfiability of the team-semantical extensions

- $\mathsf{FO}^2(\sim)$ *of two-variable first-order logic* FO^2,
- GF($\sim$) *of the guarded fragment* GF *of first-order logic,*
- ML($\sim$) *of modal logic* ML

is decidable.

In fact logspace-complete for the class $\mathrm{TIME}\left(\exp^{\mathrm{poly}(n)}(1)\right)$.

Transversals

Application:

Theorem

Every FO(∼)*-formula is equivalent to a Boolean combination of* FO*-formulas.*

Proof idea: Transversals* commute** with all Boolean operators!

Theorem

Satisfiability of the team-semantical extensions

- $\mathsf{FO}^2(\sim)$ *of two-variable first-order logic* FO^2,
- GF(∼) *of the guarded fragment* GF *of first-order logic,*
- ML(∼) *of modal logic* ML

is decidable.

In fact logspace-complete for the class $\mathrm{TIME}\left(\exp^{\mathrm{poly}(n)}(1)\right)$.

Transversals

Application:

Theorem

Every $\mathsf{FO}(\sim)$*-formula is equivalent to a Boolean combination of* FO*-formulas.*

Proof idea: Transversals* commute** with all Boolean operators!

Theorem

Satisfiability of the team-semantical extensions

- $\mathsf{FO}^2(\sim)$ *of two-variable first-order logic* FO^2,
- $\mathsf{GF}(\sim)$ *of the guarded fragment* GF *of first-order logic,*
- $\mathsf{ML}(\sim)$ *of modal logic* ML

is decidable.

In fact logspace-complete for the class $\mathrm{TIME}\left(\exp^{\mathrm{poly}(n)}(1)\right)$.

Conclusion

Transversals are a natural class of flatness-preserving team-semantical connectives, and possess a number of nice properties.

Future work:

- More classifications of team logics in the framework.
- Incorporate connectives that are not flatness preserving (e.g., temporal operators).
- Smallest unit: Atomic formulas. How to show, e.g., locality?

References

- Wilfrid Hodges. *Compositional Semantics for a Language of Imperfect Information.* Logic Journal of the IGPL 5(4), 1997, pp. 539–563.
- Bjarni Jónsson and Alfred Tarski. *Boolean algebras with operators. Part I.* American Journal of Mathematics 73(4), 1951, pp. 891–939.
- Jouko Väänänen. *Dependence logic: A New Approach to Independence Friendly Logic.* London Mathematical Society student texts 70. Cambridge University Press, 2007.